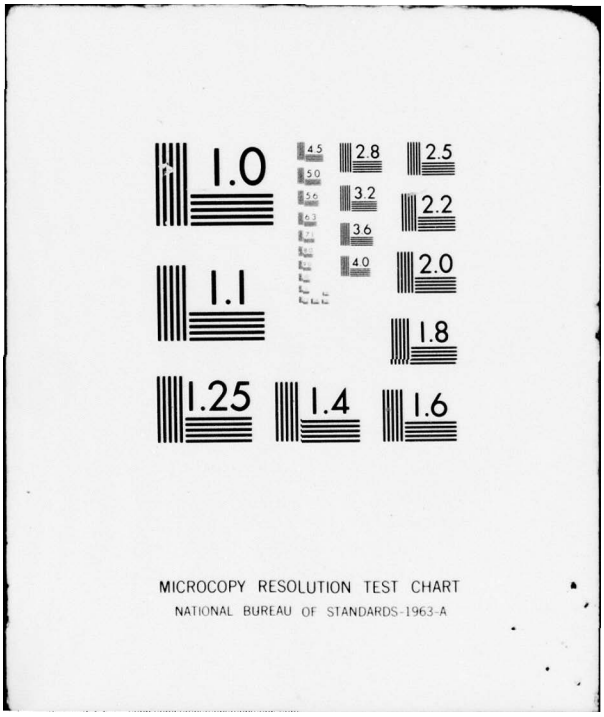
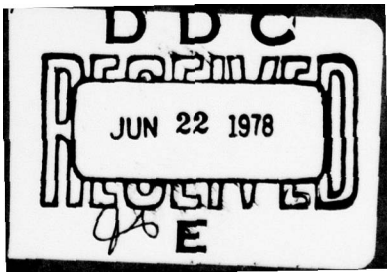
AD-A055 466 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING.(U)
MAR 78 R G RUSHE

UNCLASSIFIED AFIT/GEP/EE/78-1 NL

1 OF 2
AD
A055466


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A


DDC
JUN 22 1978
E

(1)

(6) A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING.

(9) Master's THESIS,

(14) AFIT/GEP/EE/78-1

(10) Randall G. (George) Rushe
2Lt USAF

(11) Mar 78

(12) 111p.

Approved for public release; distribution unlimited.

AFIT/GEP/EE/78-1

A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING

THESIS

Presented to the Faculty of the School of Engineering

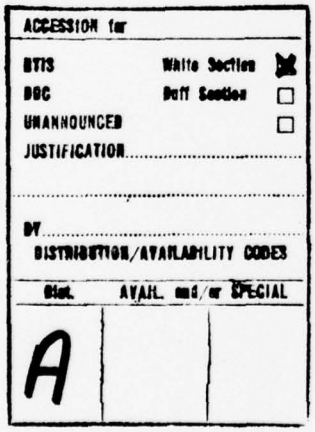
of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science



by

Randall G. Rushe, B.S.

2Lt USAF

Graduate Engineering Physics

March 1978

Preface

The design and implementation of an automated data acquisition system for the AFML Thermal Flash Test Facility is summarized in this thesis. Being primarily a physicist, I have approached this project as an application of computer techniques to a common laboratory problem--data acquisition. The computer offers immense possibilities for increasing research productivity and relieving the tedium of an often onerous chore. I have personally learned a great deal while developing this data acquisition system, and hope that it will become a useful tool for the Thermal Flash Test Facility. I also feel that many of the approaches taken will be applicable to systems of a similar nature. In addition to the discussions of system development, this report contains much description of the data acquisition system itself in an effort to insure that the user fully understands its operation and design.

Assistance from many sources was gratefully received during this project. In particular, I wish to thank Nick Olson and Ben Wilt at the Thermal Flash Test Facility for their unending support. I would also like to express my appreciation to Dr. Dale Ford, Tom Wood and Lt Dave Summer of the AFML Computer Activities office for their help and encouragement over the past several months. Dr. Gary B. Lamont, my thesis advisor, deserves special thanks for approaching me with this problem, rendering advice, and showing patience as I struggled to complete it. To my family and friends I can never adequately express my gratitude for their loving concern and moral support.

Randall G. Rushe

Table of Contents

Table of Contents (Cont.)

List of Figures

List of Tables

Abstract

Laboratory computers for automated data acquisition are becoming increasingly common. This report summarizes the development of a microcomputer-based data acquisition system for the Air Force Materials Laboratory Thermal Flash Test Facility. The system is based on a Digital Equipment Corporation LSI-11 microcomputer and acquires multiple channels of data from nuclear simulation experiments on aircraft components. Comprising the lowest level of a rudimentary hierarchial network, the system transmits the data to a larger minicomputer system for storage, from where it is subsequently retransmitted to a mainframe system for reduction and plotting. Software for the data acquisition system is modularly structured and written primarily in FORTRAN to facilitate modifications by the user. Basic hardware for the system is discussed, and the program algorithms and structure are examined.

A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING

I. Introduction

Computers have long been recognized as being invaluable for the reduction and analysis of experimental data. In the past their role has usually been limited to off-line or non real-time applications since computers were too large and expensive to be used in most laboratory environments. However, with the introduction of low cost minicomputers and microcomputers this restricted role has been expanded by making it physically possible and economically feasible to put a computer into the laboratory. The process of acquiring and storing data, as well as real-time analysis and even control of the experimental apparatus, can now be assisted by computers. Just as the hand-held calculator has relieved the scientist from much of the drudgery of long calculations, so too can the laboratory computer eliminate much of the tedium of acquiring and processing experimental data.

There are two concepts, automated data acquisition and hierarchial networks, that will be useful in understanding later discussions. Data acquisition is defined as the process of acquiring data from transducers and converting it to a form which can be processed by a computer. The primary advantage in using programmable computers for automated data acquisition is their inherent flexibility and ability to adjust to changing requirements. This flexibility through software is needed because "laboratory tasks have a natural tendency to change with time, particularly in research, so that the life of an automated system may be short unless it can be modified to cope with changing demands or adapted for alternative

applications (Ref 5:794)."

The second concept is that of a hierarchial network in which processors with varying capabilities and resources are linked together. An example of such a network is one in which the lowest element is a microcomputer dedicated to perform data acquisition for a single experiment. It may store only limited amounts of data and do little, if any, analysis. This small laboratory computer requires the next level in the network, consisting of a more sophisticated computer system, to perform extended data storage or reduction. At the highest level of the hierarchy is a very large mainframe computer system with extensive capabilities for detailed data analysis. The most important benefit of a hierarchial system in data acquisition applications is resource sharing, since each experiment at the lowest level has access to resources which are greater than those justified by a single experiment alone.

This thesis describes the development and realization of microcomputer-based data acquisition system which comprises the lowest level of a rudimentary hierarchial network. The system was constructed primarily for use in the Air Force Materials Laboratory Thermal Flash Test Facility (AFML/MBC) at Wright-Patterson AFB, Ohio. The background and initial data acquisition system specifications for the facility are described next, followed by the problem statement and an outline of the presentation.

Background and Initial System Specifications

The AFML Thermal Flash Test Facility combines a wind tunnel and creep frame with a quartz lamp bank to simulate nuclear thermal flash effects on aircraft components. It is operated by the University of Dayton under contract to the Defense Nuclear Agency (DNA) and has recently been upgraded,

one portion of which is the addition of an automated data acquisition system (Ref 14).

The previous data acquisition system for the facility consisted of a VIDAR integrating digital voltmeter, which sampled and digitized an analog signal, and a Kennedy digital tape recorder, which stored the digitized data. Handling two channels at a time, the system could acquire data at a rate of four samples per second. Later the magnetic tape containing the stored data was taken to the Aeronautical Systems Division (ASD) computer center where the data was punched on cards. These were then used as input to plotting programs on a Control Data Corporation (CDC) 6600 computer and output was routed to a remote for plotting. A block diagram of this original data acquisition system is shown in Fig. 1.

Basic requirements for an automated data acquisition system initially called for one capable of handling six channels of input with 0-100 millivolt signals, sampled at rates of up to ten samples per second. All samples were to be taken in a strictly time-based mode; that is, at fixed intervals rather than dependent upon signal values or test events. Tests were to be of up to 100 seconds duration, and the final output required was high quality report graphs. Three data acquisition system approaches were considered by AFML to fulfill these basic requirements. The first consisted of using a data transmitter to send data over a 9600 baud dedicated line to the AFML Systems Engineering Laboratories (SEL) Model 86 minicomputer which would control the acquisition of data directly. The second approach would be implemented using a Control Logic 8080 microprocessor to acquire data and transmit it over a 300 baud telephone line to either the ASD CDC 6600 or AFML SEL 86 computers for reduction. The final approach to be considered used a Digital Equipment Corporation (DEC) LSI-11 microcomputer to control

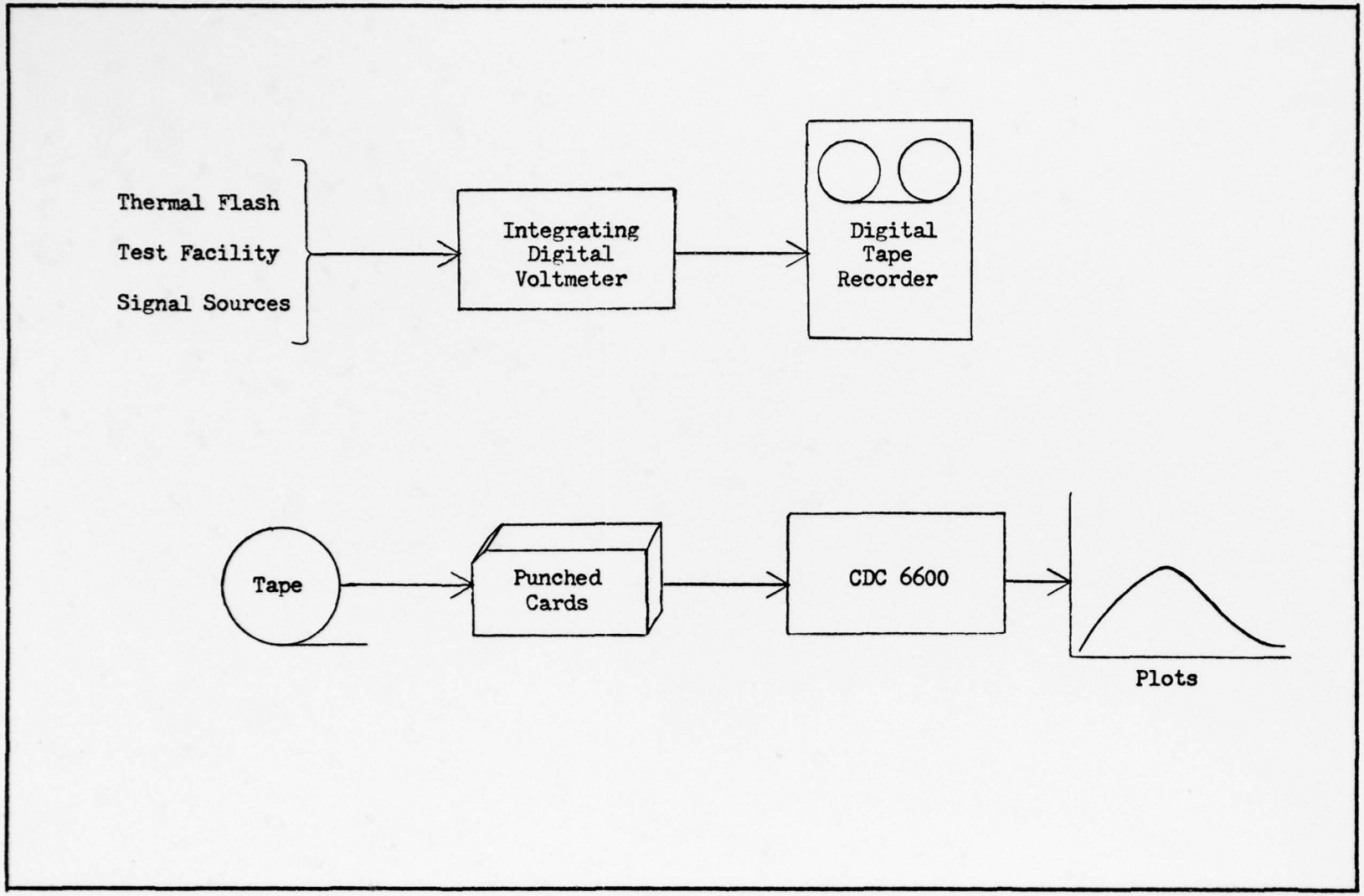


Figure 1. Former Thermal Flash Test Facility
Data Acquisition System

data acquisition, store the data, and then transmit it over a 300 baud telephone line to a larger computer.

The DEC LSI-11 based system was selected to implement the data acquisition system because of a number of factors. Among these is the availability of a software development system and FORTRAN complier for the LSI-11, which simplifies the task of program development. Also, there is a large selection of readily available modules and peripherals for use with the LSI-11. This system is able to acquire and process data independently of the SEL 86 and CDC 6600 which makes it preferable over the data transmitter implementation, and the speed and 16-bit architecture of the LSI-11 is an advantage over the Control Logic 8080 based system.

The task of building and implementing the data acquisition system was submitted to the Air Force Institute of Technology (AFIT) as a master's degree thesis proposal. The problem statement follows:

Problem Statement

The objective of this investigation is to implement, using the DEC LSI-11 microcomputer as the basis, a data acquisition system which will acquire data from the AFML Thermal Flash Test Facility, and transmit it over a 300 baud telephone line to the SEL 86 minicomputer system for storage. From there the data will be sent over a data line to the CDC 6600 for analysis and graphic output in the form of plots. The implementation of the system includes the assembly and interconnection of required hardware, as well as the design and development of appropriate LSI-11 software.

Outline

Chapter II contains the basic design requirements for the data

acquisition system and states assumptions made for its implementation. Overall system, hardware, and software organization are discussed in Chapter III. Chapter IV describes the hardware which comprises the data acquisition system, but detailed documentation is included in the appendices. The software developed for the LSI-11 to implement the data acquisition system is the subject of Chapter V, which discusses the basic algorithms and software design considerations. Actual source code is contained in a companion volume to this thesis (Ref. 11). Some conclusions are presented in Chapter VI, along with recommendations for future system improvements. Appendix A contains detailed hardware information such as hardware module jumper selections, and pin assignments.

II. System Requirements and Assumptions

Introduction

The intent of this chapter is to state the system requirements for the data acquisition system, and to list implementation assumptions. These specifications and assumptions were derived from the thesis proposal submitted to AFIT; and from consultations with the user and the AFML Computer Activities (AFML/DOC) office, who provided support for the project. Functional system requirements are included in the discussion of system specifications, as well as software constraints and documentation needs. Assumptions deal with the system environment, hardware, and software.

Data Acquisition System Specifications

Now that the scope of the data acquisition problem and general objectives are known, as given by the problem statement in Chapter I, specific requirements for the data acquisition system will be defined in this section. The functional system requirements are described first, followed by a discussion of software constraints. Finally, some documentation requirements will be presented.

Functional System Requirements. There are a number of operational demands which must be met by any automated data acquisition system developed for this project. These functional system requirements encompass the actual data acquisition task, the management of system parameters, and the transmission of data to other computers in the network. Also included is the limited on-line data reduction and time-sharing terminal emulation required of the system. The specific requirements for each of these functions is discussed next.

1. Data Acquisition Task. The most basic needs which must be

satisfied by the data acquisition system concern the actual acquisition of data from the experimental apparatus. These dictate that the system have the following capabilities, summarized in Table I, as specified by the user:

a. Up to eight channels of input must be allowed. Although most thermal flash tests presently run require only one or two data channels, the system must be able to accommodate plans for future expansion which call for additional signal inputs.

b. Signals must be accepted over a broad range of o-1000 millivolts, and varying signal ranges must be permitted for each channel independently. This will allow several different transducers to be used as signal sources. Types of signal sources and their characteristics are discussed fully in Chapter IV.

c. Data must be sampled on active channels in a time-based mode at rates as high as 100 samples per second, and each channel must be able to be sampled independently of the others at rates assigned by the user. This enables only as much data to be taken from an individual channel as is required for a given test.

d. Acquisition of data must commence either on command, or automatically by synchronization with an external start signal. Synchronization with a signal generated by the test facility control equipment allows data to be taken only during the portion of an experimental run which is generating valid data. The manual start capability is required if no synchronization signal is present, or for test purposes.

e. Data must be digitized and stored with a resolution of 12-bits--one part in 4096. This is the resolution available from the

analog-to-digital (A/D) converter used in the system, which is described in Chapter IV, and is sufficient resolution for the user's data analysis and plotting routines.

f. A time base must be recorded with each data point which is accurate to 0.01 seconds and which is measured from the start of data acquisition. Corresponding to the maximum sampling rate, this accuracy is more than sufficient for the user's present needs and will handle increased system demands.

g. Test runs from 0-300 seconds duration must be handled by the data acquisition system, and will vary according to experimental conditions.

Table I

Data Acquisition Requirements

Number of input channels	8
Input signal range	0-1000 mV
Sample rate	0-100 samples/second
Acquisition start	Manual or automatic
Data resolution	One part in 4096
Time base accuracy	0.01 seconds
Test duration	0-300 seconds

2. Management of System Parameters. Several parameters which govern the operation of various system functions must be handled by the data acquisition system. These system parameters will be divided into three general classifications for discussion: a) test parameters controlling data acquisition which may or may not vary from run to run, b) acquisition run parameters which are unique to each test run, and c) user-defined plot parameters which are passed with the data to the analysis and plotting

routines. The values of each of these parameters must be easily changed by the user in order to adjust to changing system requirements or system configuration. They are discussed below.

a. Test Parameters. The test parameters control the acquisition of data during each run, and are to have default values which are loaded with the system program. New values must be able to be input at any time between runs, and current values must be printed when desired. The test parameters are listed here:

(1) The sampling rates for each channel determine which channels are sampled and how often.

(2) The signal gain for each channel indicates the amplification of the input signal, and allows data to be converted to reflect true signal magnitude.

(3) For automatic acquisition the number of the channel used to synchronize data acquisition with an external start signal, the voltage used as the synchronization threshold, and the gain of the synchronization signal are required.

(4) The sampling delay is the length of time sampling continues after the end of a test run, and combined with the run time discussed later it determines the total data sampling duration.

b. Acquisition Run Parameters. At the beginning of each test run there are three additional parameters which must be input by the data acquisition system. Together with the test parameters they control the acquisition and handling of data

for the run. No default values are to be allowed for these parameters which are described below.

(1) An unique five character run identifier is used to establish a label for the data taken during a run, and is the file name under which the data will be stored as a SEL 86 permanent disk file.

(2) A sixty character run title further identifies and describes a set of test data.

(3) The run time indicates the duration of sample exposure, and with the sampling delay determines the length of time data is sampled.

c. Plot Parameters. Plots are part of the final output of the data acquisition system, and are generated by user programs on the CDC 6600. In order to govern the operation of the plotting routines, a set of parameters associated with each data channel must be passed up the network from the LSI-11. As with the test parameters, these must have values which are loaded as defaults and which may be changed or displayed on command. The plot parameters are:

(1) Single plot codes for each channel to determine the type of plot to be generated.

(2) Twenty user-defined special codes for each channel used to pass miscellaneous information such as conversion constants or plot scale factors.

3. Data Transmission. After data has been acquired and temporarily stored by the LSI-11 during a test run, it must then be transmitted along with pertinent test, plot, and acquisition run parameters to the SEL 86

for permanent storage. The data acquisition system must then initiate subsequent retransmission of the data and parameters from the SEL 86 to the CDC 6600 for reduction and plotting. The transmission of data may be either performed automatically after each test run, or on command by the operator. Human intervention required must be held to a minimum, preferably limited to dialing up the SEL 86 prior to transmission and hanging up afterwards.

4. On-Line Data Reduction. The data acquisition system must also be capable of performing a limited amount of data reduction at the LSI-11 by calculating and printing peak data information for each test. This includes, for each channel, the peak signal level in millivolts and the time in seconds after the start of acquisition that the signal occurred. The information gives the technician immediate feedback after a test so that the validity of the run or calibration of the equipment can be checked, and so that a decision can be made whether to transmit the data for storage and plotting. Print out of the peak information must be performed automatically after each test or on command.

5. Time-Sharing Terminal Mode. Another requirement of the data acquisition system is that a facility must be provided to allow the user to communicate directly with the time-sharing systems of the SEL 86 and CDC 6600. This capability is needed so data or control information in files stored on the SEL 86 may be edited, or so program changes can be made on the CDC 6600, without requiring a separate time-sharing terminal. By enabling the system to emulate a terminal there will be a transparent interface between the LSI-11 user and the dial-in telephone lines of another computer system.

Software Constraints. In addition to the functional requirements

demanded of the data acquisition system, there are some general software restrictions imposed by user needs. The first of these is that the program which implements the system must be general enough to handle changing test requirements without necessitating major revisions of code. Apart from run-to-run changes in the system parameters discussed earlier, which must be handled as a matter of course, there exists a possibility of future expansion of the system in order to meet additional test demands, or of use in other experimental facilities. Program organization has to be made with these factors in mind.

Eventually changes will have to be made, and this raises the second software requirement--ease of software maintenance and modification. The user may find his software support rather limited in the future, hence the system programs must be well structured and written so as to make later changes as straightforward and simple as possible. A corollary to this is the demand that extensive use be made of a higher-order-language to facilitate program modifications by the user. For this system FORTRAN IV is used since a compiler is available for the LSI-11, and because personnel at the laboratory are familiar with the language. However, FORTRAN does not expedite structured programming. Portions of code which cannot be written in FORTRAN, or which are time-critical, may still be written in assembly language. This will include the drivers and interrupt service routines for the system clock, analog-to-digital converter, and modem. Additionally, these assembly language routines will be unaffected by minor changes in system configuration.

Another demand made on the software is that the operation of the data acquisition system must be simple from the user's viewpoint, and as "idiot-proof" as is consistent with other system requirements. The first point

implies that system set-up should be easy, commands should be meaningful and concise, and that the operator should be prompted whenever input is needed. The second point requires that the system detect and point out errors in user input, and not allow the limits of the system to be exceeded (for example, specifying a non-existent channel or requesting too high a sampling rate). This should be true not only for the total finished system, but also for those assembly language routines and sections of FORTRAN code likely to be used individually or altered in the future.

It is also desirable to maintain program size within the 8K words of memory initially available for the system. However, this may be overridden by other factors just discussed since additional memory can be installed up to 28K.

Documentation Requirements. Hardware configuration and software for the data acquisition system must be fully documented to explain system operation, and to expedite future system modifications. The hardware documentation will include the interconnections, options chosen, and pin assignments which are made, so that information is provided for system reconfiguration and maintenance.

Software documentation must be thorough; and will include descriptions of each major algorithm, as well as discussions of program structure and strategy. Source listings with comments must also be provided, although they are not included as an integral part of this thesis. Complete documentation will enable system programs to be modified later with little difficulty, and will give the user an explanation of system operation.

Assumptions

Several assumptions will be made for the development of the data

acquisition system. Dealing with system environment, hardware, and software these assumptions are stated here:

1. The sampling rates and digitization resolution decided upon by the user are sufficient to assure adequate representation of the input signals.
2. Noise levels in the system are negligible with respect to the expected signal levels.
3. The major hardware components for the data acquisition system have been purchased but not installed. They were selected on the basis of preliminary design discussions between AFML/DOC and AFML/MBC, and will be used for this system. Specific design criteria are discussed in Chapter IV. These hardware components are: the LSI-11 processor, 8K words of memroy, and backplane; 1000 Hz crystal controlled clock; 12-bit analog-to-digital converter; teletype and serial interface; differential amplifiers; and a +5V, +12V power supply.
4. For purposes of system development, amplifiers and the analog-to-digital converter are considered to be calibrated close to optimal values.
5. The Thermal Flash Test Facility control system will generate a signal at the beginning of each test which marks the start of automatic data acquisition.
6. A disk-based DEC operating system (RT-11) is available for LSI-11 software development, and will have an assembler, FORTRAN IV compiler, and text editor.
7. A stand-alone RT-11 simulator is available for use in the data acquisition system. This performs some of the functions of the

RT-11 operating system and enables FORTRAN routines to be used in the LSI-11 system program; it is discussed in Chapter V.

8. Programs are loaded into the data acquisition system from punched paper tape which is in DEC standard absolute loader format. Paper tape will be used as the object code medium since the data acquisition system teletype has a tape reader, and because the development system can produce loader format binary tapes.
9. Sufficient LSI-11 memory will be available to implement the data acquisition system programs by allowing expansion of the initial 8K words installed.

Summary

The data acquisition system specifications and assumptions, based on consultations with the user and others, have been discussed in this chapter. Together they determine the design criteria for the system, whose overall organization is presented in Chapter III.

III. Data Acquisition System Organization

Introduction

This chapter examines the organization of the automated data acquisition system developed to fulfill the requirements and specifications discussed in Chapter II. In addition to those user-defined requirements, design of the system is affected by the available hardware and software resources, and by the system environment. Together, these considerations determine the nature of the data acquired, how it is processed, the configuration of the LSI-11 system, the implementation of the software, the form of communication with other computers in the network, and the interaction of the system with the user. All of these factors have either already been stated as assumptions in the last chapter, or they will be discussed later. However, in order to present a coherent picture an overall view of the general organization of the system implemented will be taken first. This will show how the system functions, both from the standpoint of signal and data flow, and from the user's operational viewpoint. Then there will be a discussion of the system hardware and software organization; although individual elements of each are presented in greater detail in Chapters IV and V, respectively.

General Organization

The objective of the data acquisition system, as stated in Chapter I, is to acquire data from tests run at the Thermal Flash Test Facility and to transmit the data for reduction and plotting. A block diagram of the system layout used to achieve this is pictured in Fig. 2. Signals from test facility equipment are conditioned and input to an analog-to-digital (A/D) converter. There they are sampled and digitized under control of the

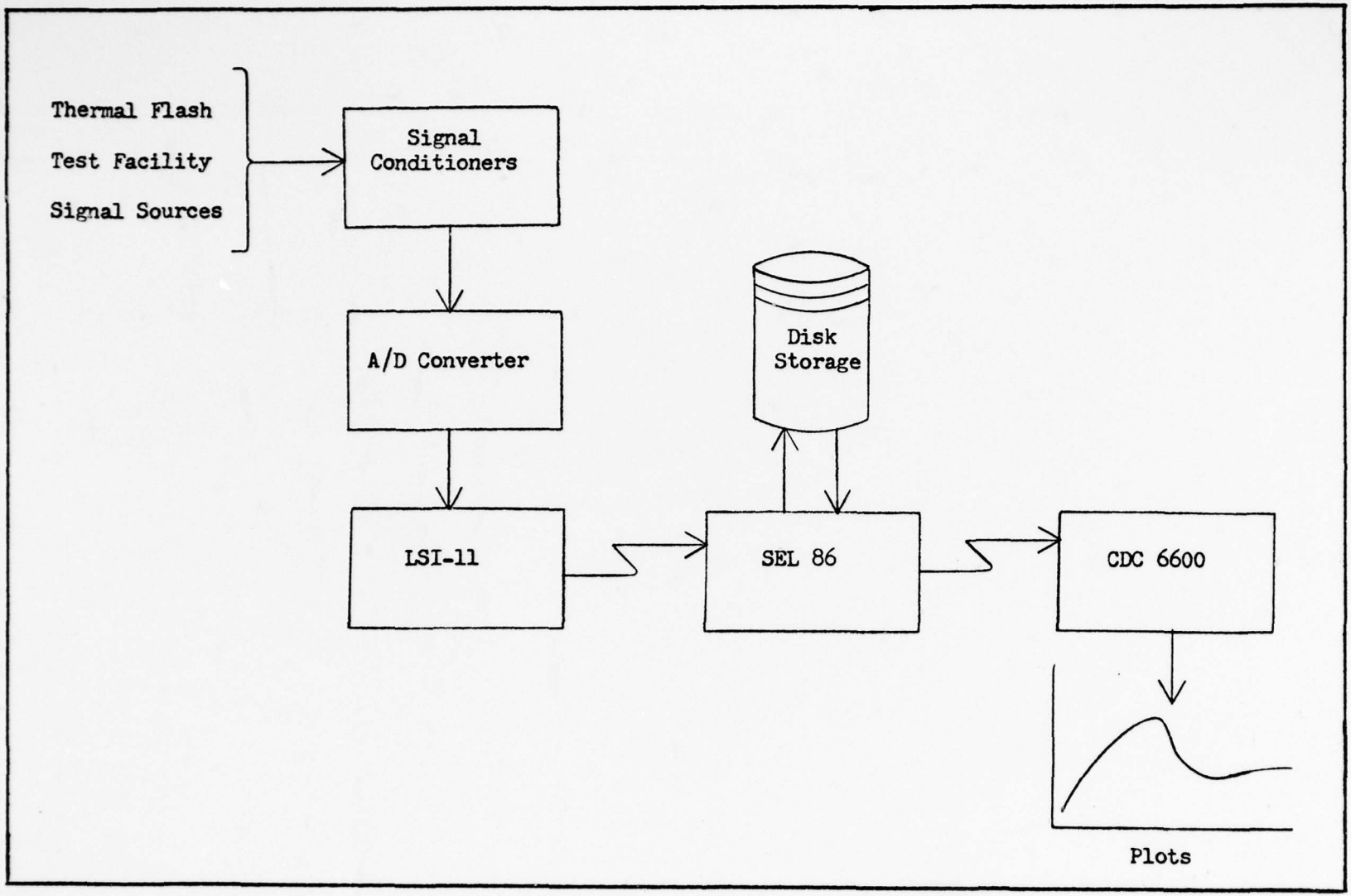


Figure 2. Data Acquisition System Block Diagram

LSI-11 which temporarily stores the converted data in memory. At the conclusion of a test any requested on-line data reduction is done; then the data is transmitted by a modem over a 300-baud telephone line to the AFML SEL 86 computer system. There it is permanently stored on disk and, after being appended with the necessary job control language (JCL), the data is submitted by the SEL 86 as batch job input to the ASD 6600. User reduction and plotting routines which are resident on that system input the data and generate the desired plots. These are routed to one of the ASD computer system remotes for output.

Operation of the data acquisition system from the user's viewpoint is shown in Fig. 3. The material sample to be tested is mounted in front of a quartz lamp bank and, depending on the tests being run, may also be positioned in a wind tunnel or static-load creep frame. The required signal transducers are set up, some of which are attached to the sample, and signal cables are connected to the signal conditioning equipment. The gains of the signal conditioning amplifiers are selected to bring the expected signal levels within the range of the A/D converter. Now, after loading the data acquisition system program, the technician must set the system test and plot parameters discussed in the last chapter to control data sampling and plot generation. He then initiates data acquisition by typing the appropriate command and then entering the final set of acquisition run parameters. If automatic acquisition start has been specified in the test parameters, the system will wait until the signal on the channel used for test synchronization crosses a predetermined threshold before data sampling begins. For a manual start the operator hits the RETURN key when sampling is to commence. The technician begins the thermal flash test run using the thermal flash control equipment. After the test run is

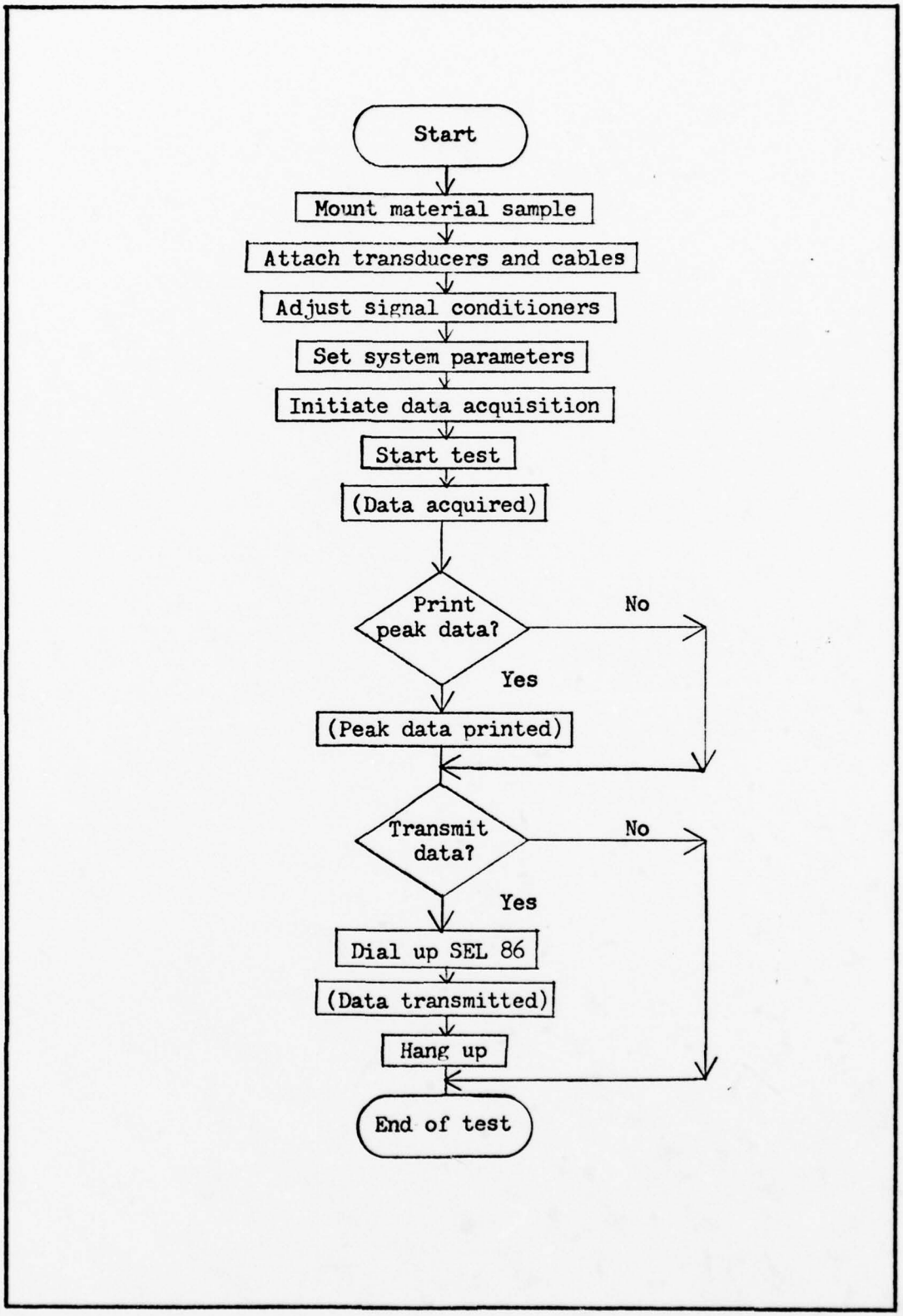


Figure 3. Material Testing Sequence

complete, peak data values for each channel and the times at which they occurred are printed on the teletype if desired. Then if the data is to be transmitted for storage and plotting the operator dials up the SEL 86 computer and the data acquisition system will begin transmission. When the system has finished transmitting data the operator hangs up the phone and repeats the preceding procedure for the next test. Meanwhile, the SEL 86 retransmits the data and parameters to the CDC 6600 which produces the required plots.

Hardware Organization

Now that a general view has been taken of the data acquisition system and its operation, the hardware organization of the system implementation will be briefly described. Any detailed discussion of specific hardware elements or design considerations will be deferred until Chapter IV. The general layout of the hardware components comprising the data acquisition system is diagrammed in Fig. 4.

Signals input to the system are derived from thermal flash facility transducers. These transducers are selected to measure particular physical characteristics of the material being tested, such as temperature and expansion, or to monitor test conditions; for example, quartz lamp flux. The signals from the transducers normally require some form of conditioning, and are transmitted through cables to the signal conditioning equipment. Amplifiers with variable gain are used to increase low-level transducer signals so they are near the maximum range of the A/D converter, and hence will have greater resolution when digitized. Additionally, signals from thermocouple transducers are passed through temperature compensation circuitry built into the amplifiers to eliminate the need for

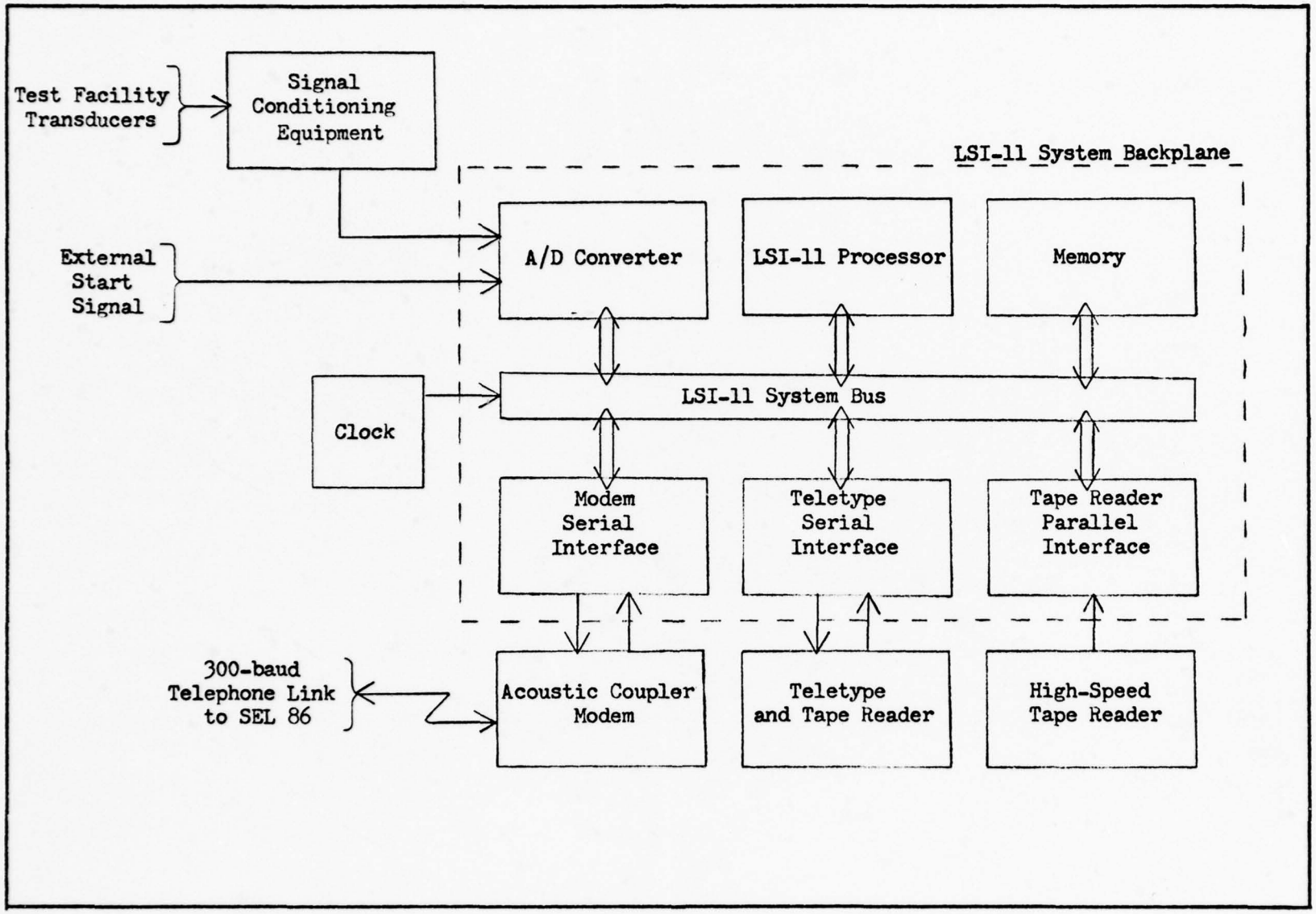


Figure 4. Data Acquisition System Hardware

reference junctions. The conditioned signals are routed by cable to the inputs of the A/D converter. A non-data signal also input to the A/D converter is the external acquisition start signal generated by test facility control equipment.

Plugged into the LSI-11 system backplane, the A/D converter is directly connected to the LSI-11 bus. On command from the LSI-11 it selects one of the input channels, samples and holds the voltage level present, and digitizes the signal. The LSI-11 processor, also mounted in the backplane, communicates with memory modules and other devices through the system bus. An external clock is used to facilitate timing of data sampling and other system functions. The signal it generates enters LSI-11 through a special interrupt line in the bus.

Communications with the human operator is accomplished using a standard teletype which is controlled by a serial interface module inserted into the system backplane. The teletype also features a paper tape reader which is used to load the data acquisition system program. A high speed paper tape reader with a parallel interface was also used during software development. To communicate with other computers in the network, either during data transmission to the SEL-86 or when emulating a time-sharing terminal, an acoustic coupler modem is used to interface the system with a telephone line. The modem is linked to the system bus through another serial interface module.

Software Organization

The organization of data acquisition system software is described in this section, and includes discussions of program strategy and structure. As with the section on hardware organization, details concerning the design

and realization of individual algorithms will be examined in a later chapter.

In keeping with the system specifications outlined in Chapter II concerning software constraints, namely generality and ease of modification and maintenance, a modular approach to the design and implementation of system software is followed and extensive use is made of FORTRAN. Presented below is the basic organization of the major data acquisition system FORTRAN subprograms, followed by an examination of the assembly language routines. There is then a short discussion of system data structures.

Major FORTRAN Routines. The main portion of the data acquisition system is made up of modular FORTRAN subprograms which perform well-defined functions. However, a strictly structured approach to the design of these modules was tempered somewhat by a desire to decrease system overhead by keeping down the number of subroutine linkages. Hence, a few of the routines perform more than one major operation, but good structure is still maintained.

Organization of the primary FORTRAN programs is depicted in the block diagram of Fig. 5. These subroutines perform the following functions in order to fulfill the system requirements stated in the last chapter: accept commands typed in by the operator, manage test and plot parameters, input acquisition run parameters, carry out the actual data acquisition task, allow for the automatic calculation of peak data and automatic transmission of data, calculate and display peak data information, transmit data to the SEL-86, and emulate a time-sharing terminal.

The main program is a command input decoder which accepts commands typed in by the operator and then calls the appropriate subprogram to perform the requested function. This construction gives the technician great flexibility in determining which operations are to be done by the system.

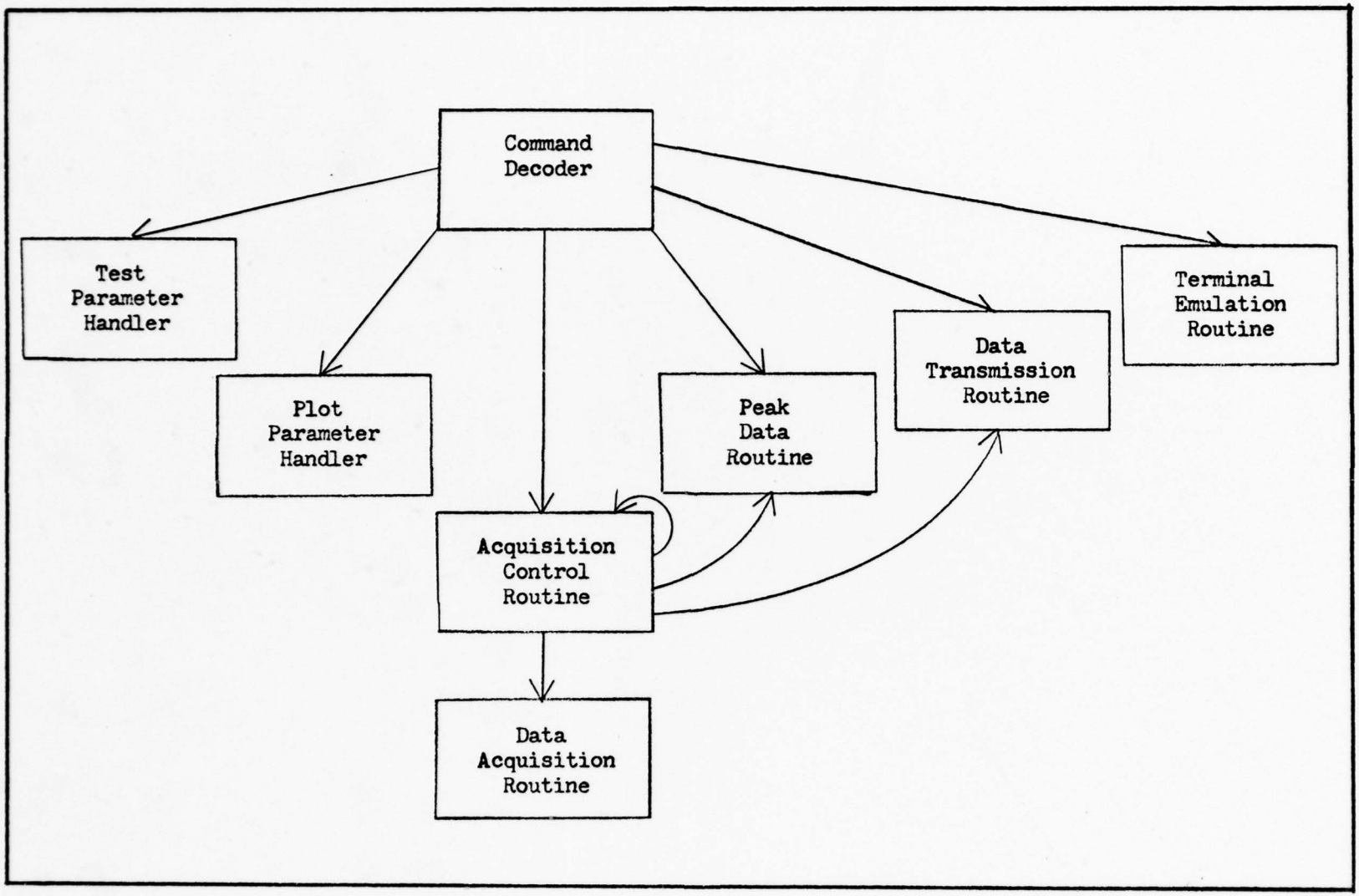


Figure 5. Primary Data Acquisition System FORTRAN Routines

It also enables additional functions to be handled by the data acquisition system merely by adding the required subroutines and then modifying the decoder to recognize additional commands. The subroutines presently included in the system perform the operations discussed in the Chapter II section on functional system requirements.

The test parameter handler allows the technician to make changes in the test control parameters or to print the values presently in effect. Likewise, the plot parameter routine displays the user-defined plot codes or permits changes to be made by the operator. Both of these routines check user input for validity, as do the command decoder and the acquisition control routine, discussed next.

The data acquisition control subroutine performs a number of operations. First, it inputs the acquisition run parameters from the technician prior to a test, and then calls the data acquisition routine to perform the actual data acquisition task. On return from that routine, after data has been acquired and stored, a call is made to the peak data calculation and data transmission routines which conditionally perform their respective functions if they are to be done automatically after a test. If automatic re-acquisition has been enabled, the acquisition control routine immediately requests a new set of run parameters for the next test. Otherwise it returns control to the command input decoder.

Collection of data is accomplished by the data acquisition routine, called from the acquisition control subroutine. It starts acquisition, either manually or by automatic synchronization, and then samples data at the required intervals. The data taken and a time base reference for each point are stored in memory.

The peak data routine determines the maximum signal value which

occurred on each active channel during a test run, and prints it along with the time at which the peak was reached. This routine also processes conditional requests to display the peak data information when called from the acquisition control subroutine.

Transmission of data and parameters to the SEL 86 for storage and retransmission to the CDC 6600 is done by the data transmission routine. It prompts the operator to dial up the SEL 86 and establishes communication with that system. After all the data has been transmitted the routine tells the SEL 86 to send the data with appropriate JCL to the CDC 6600 for any plots to be generated. Then the transmission subroutine logs out from the SEL 86 and prompts the technician to hang up. Like the peak data routine, the transmission routine handles conditional requests for data transmission.

The final functional requirement outlined in the last chapter was that the data acquisition system must be able to operate transparently as a time-sharing terminal. This function is done by the terminal emulation routine.

Assembly Language Routines. The data acquisition system routines written in assembly language perform operations in support of the FORTRAN subroutine tasks. Their functions are: 1) service the anglog-to-digital (A/D) converter to permit the LSI-11 to acquire data and process it in an efficient format, 2) service the system clock and provide software timers so that an accurate time base is available and so that data can be taken at the required rates, and 3) service the modem interface to allow communication with other computers from the FORTRAN routines. As depicted in Figs. 6a, 6b, and 6c, to maintain a modular structure there are actually several routines that perform particular functions within each of these areas. The operation of each of these will be briefly described below; but, again,

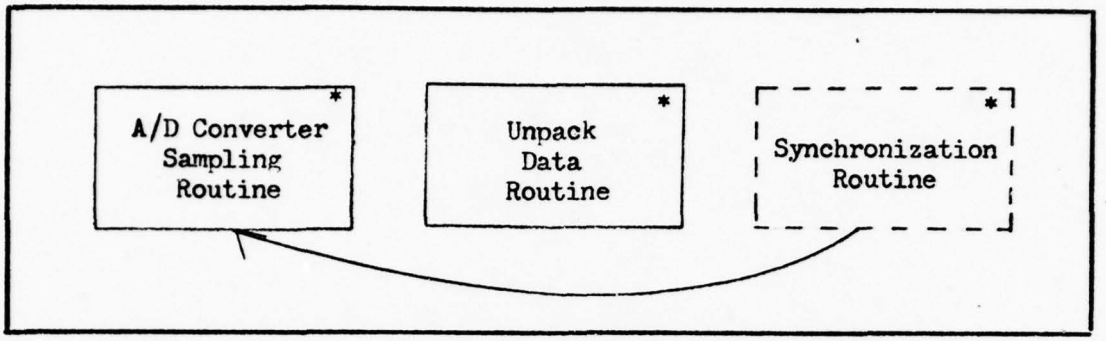


Figure 6a. A/D Converter Assembly Language Routines

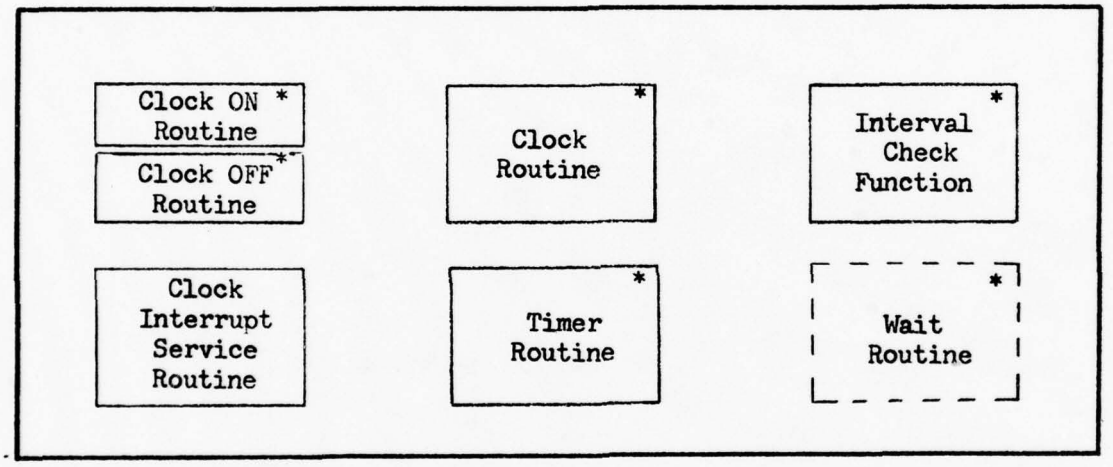
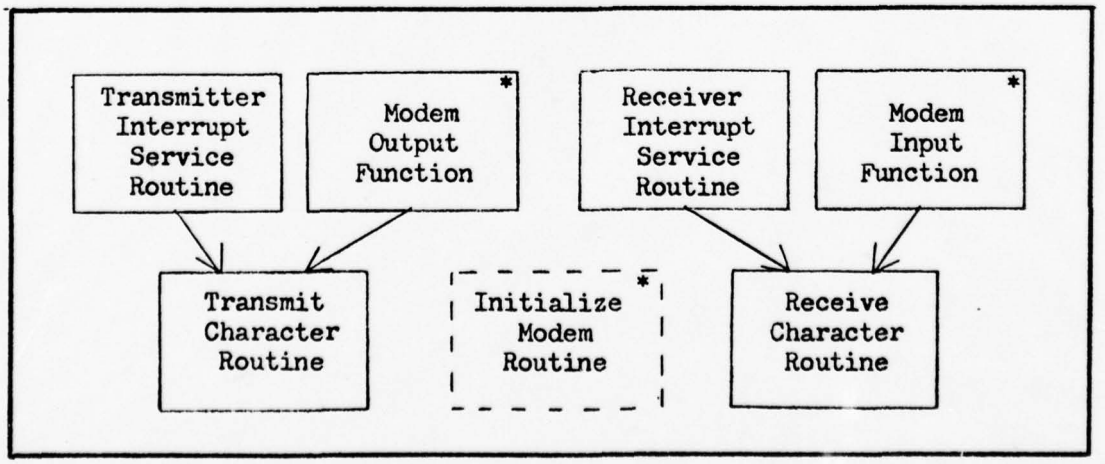


Figure 6b. System Clock Assembly Language Routines



*called from FORTRAN

Figure 6c. Modem Assembly Language Routines

details are left to Chapter V.

It should be noted here that all of the assembly language routines are designed to be useful tools not only in the present implementation of the data acquisition system, but also in other systems which may be developed by the user. Therefore, those routines that are called from FORTRAN code follow the compiler's conventions for subroutine linkage (Ref 9:2-3 to 2-5). They also check for valid argument lists and use the FORTRAN error handler to trap errors. Additionally, there are some routines included that are not used in the present system, but which are available for future use if the need arises.

1. A/D Converter Routines. The A/D converter sampling routine causes the A/D converter to sample the desired channel; then returns the digitized data, the time at which the sample was taken, and the channel number packed with the data. To unpack the compact channel and data format (see the section on data structure) the unpack routine is called. The synchronization routine, not currently used, continually calls the sampling routine to sample the signal on a given channel, waits until it exceeds a certain level, and then returns.
2. System Clock Routines. The collection of routines associated with the system clock service interrupts, control the operation of the clock, and perform functions in connection with software timer operation. The interrupt routine increments a clock count in hundredths of seconds, and services the system timers. The clock routine reads and sets the system clock, and the clock switch routines allow the clock to be turned on and off. The timer routine is used to load new interval values for the software

interval timers, or to read present timer values. To determine if the time interval for a certain timer has elapsed, the interval check function is called. Not used by the data acquisition system is the wait routine which waits until a given time elapses before returning.

3. Modem Routines. Communication through the acoustic coupler modem is achieved using the modem routines. The transmit character routine is used to transmit a single character from an output character from an output character buffer, and is called by the transmitter interrupt service routine to send another character. The modem output function is called from FORTRAN to place a character into the output buffer, after which it calls the transmit character routine to attempt to send a character. It is also used to see if the modem is ready by checking for data-set-not-ready errors. Performing similar functions for the modem receiver are the receive character routine, the receiver interrupt service routine, and the modem input function. The initialize modem function re-initializes the modem input and output buffers, but is not presently used.

Data Structure. Inasmuch as it affects the design and organization of system software, some comments concerning data structure are in order. Specific details will, however, be discussed in Chapter V.

Extensive use is made of COMMON blocks rather than long argument lists for passing data between subroutines. This is done in order to decrease system memory overhead since the FORTRAN compiler used handles COMMON variables as efficiently as local variables (Ref 9:4-2). Employing COMMON blocks also enables the use of a BLOCK DATA section which will be discussed momentarily. The strategy for variable usage generally followed is to use

COMMON storage for variables shared among more than two subroutines. Variables used within a single subprogram are local, and those shared between only two routines are usually passed as subroutine arguments. The last case also includes control variables, function arguments, and error flags.

An interesting feature of the data acquisition system program is the use of a BLOCK data section to control the software configuration of the entire system. It sets the default values for all the user-definable system parameters and assigns values to system constants. These constants include array bounds, conversion constants, and system configuration data such as the number of channels or clock frequency. All system FORTRAN routines reference these variables through COMMON blocks and function accordingly. Hence, practically any change in the parameter default values or system configuration can be made not by changing program code, but by altering the appropriate variables in the BLOCK DATA section.

To increase system flexibility arrays are declared in the FORTRAN subroutines has having a single dimension of length one. The true array bounds are then stored in other variables whose values are defined by the BLOCK DATA section and accessed by the subroutine. For arrays having more than one dimension an arithmetic function is used to perform the correct mapping.

To save on storage requirements and also increase flexibility, the A/D converter data is stored in a compacted format which also contains the channel number. As shown in Fig. 7, the four most significant bits of a 16-bit integer word are used to store the number of the channel which was sampled, and the remaining twelve bits contain the digitized data generated by the A/D converter. The 12-bit data must be multiplied by conversion factors to obtain the true value of the signal. This format allows integer

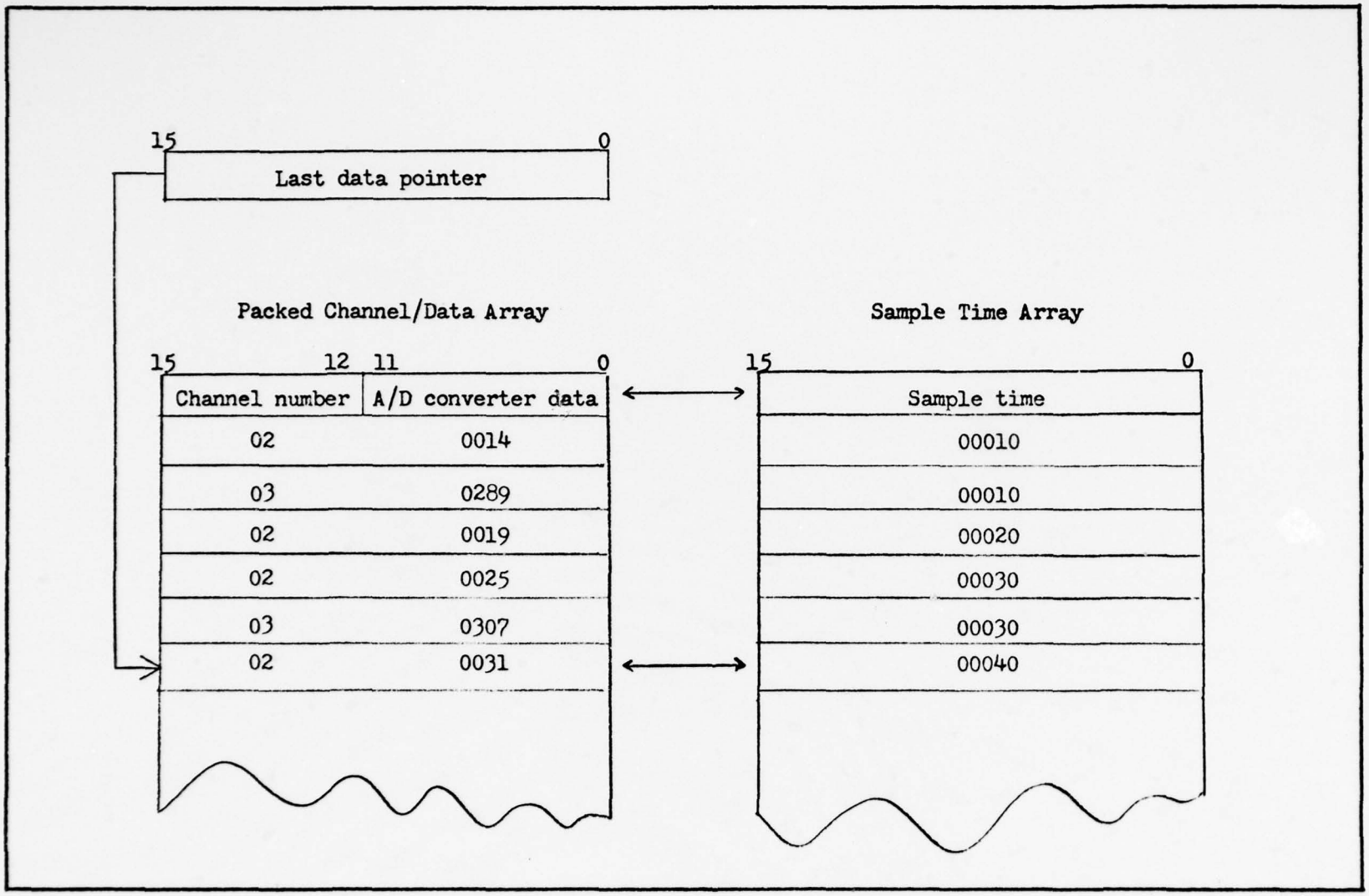


Figure 7. A/D Converter Data and Sample Time Array Format

storage to be used rather than the 32-bit floating point format. Also, by packing the channel number with the data, a single one-dimensional array can be used to store all A/D converter output. With each data point associated with a sample time stored in a parallel integer array, the order or rate of channel sampling is completely flexible. A pointer is used to indicate the end of valid data in both arrays. The packed data format is generated by the A/D converter sampling routine, and can be unpacked by the unpack data routine, discussed earlier.

Other data structures which do not have a widespread effect on system software organization will be described in Chapter V.

Summary

The organization of the data acquisition system based on the specifications and assumptions described in Chapter II has been presented. The general system organization was examined as well as hardware and software organization. Specifics for both of these areas will follow in the next two chapters.

IV. Data Acquisition System Hardware

Introduction

The successful development of a data acquisition system requires a detailed comprehension of the system hardware involved; not only because an understanding is needed of the origins of data signals and the changes which they undergo, but also because in a specialized and dedicated computer system there is a great deal of dependence on hardware device characteristics. Additionally, the development of this data acquisition system required the assembly and interconnection of system hardware, as set forth in the problem statement.

The purpose of this chapter is to describe the hardware elements comprising the data acquisition system, whose overall organization was discussed in Chapter III, and to state some of the hardware design considerations. The discussion of system hardware which follows will be divided into these areas: test facility signal sources, signal conditioners, the analog-to-digital converter, and the LSI-11 microcomputer and associated equipment. Appendix A contains detailed hardware documentation for pin assignments, and hardware module configurations.

Signal Sources

The basic elements of any data acquisition system are the sensors, or transducers, which change some physical parameter into electrical analog signals. The characteristics of these signal sources determine, to some extent, the requirements of other system hardware.

For the Thermal Flash Test Facility the primary signal sources are thermocouples, used to measure the temperature of various areas of a material sample during a test. These devices generate a voltage that is a

function of the temperature difference between a reference junction of two metal alloys and a junction in contact with an object whose temperature is being measured. Several thermocouple alloys may be employed, but the one generally used with this system is chromel-alumel. It produces signals of 0.80 mV to 9.75 mV for temperatures of 20°C to 250°C, and a 0°C reference. This temperature range is typical for the thermal flash tests that are run.

Another type of transducer used is the extensometer which produces a change in its output signal that is proportional to mechanical extension or compression. This data is taken from a material sample put under a constant load in a creep frame while being subjected to a thermal flash. Typical changes in extensometer output are on the order of 10 mV.

Radiometers are sometimes employed in order to measure the heat flux from a test facility quartz lamp bank during a test. These generate signals of a few tens of millivolts in response to the radiative flux incident on the detector.

It is also appropriate to discuss in this section the external data acquisition start, or synchronization, signal. Although not part of any useful data gathered by the system, it is produced by the thermal flash control equipment to indicate when automatic sampling of data is to begin. This allows data acquisition to be started at some predetermined point of the material test sequence. If less than the maximum number of channels for the system are being used for data, the synchronization signal may be input to one of the unused channels. However, if there are no free channels the synchronization signal must be placed onto one of the data lines. To prevent interference with the data signal in this case, the synchronization signal line is not directly coupled to the data line but is only

momentarily switched onto it through a relay when the start signal is generated.

A large variety of other signals may be used as inputs to the data acquisition system, up to a system total of eight at any one time. Proposed for future use with the system are signals to monitor other test parameters, such as lamp current or voltage. Some signal sources presently used or proposed for use with the data acquisition system are shown, as they relate to the test facility experimental apparatus, in the diagram of Fig. 8.

The signals derived from the sources just examined usually require some form of conditioning which is discussed in the next section. However, to reach the signal conditioning equipment they must pass through some type of cable. For low-level signals this cabling is particularly critical since it must also shield against noise in a laboratory environment of high voltages and large currents. To combat this noise problem shielded cable is used whose shield is grounded, with the low side of the source, at one end. The shield is connected to a conditioning equipment guard input at the other end (Refs. 4,7).

Signal Conditioners

Signal conditioning is the modification of signals entering the data acquisition system. The purpose of signal conditioning for this system is to change the input signals from the signal sources to a form that is compatible with the characteristics of the A/D converter, discussed in a moment. This signal conditioning is accomplished primarily by amplifying the low-level signals characteristic of most of the signal transducers so that they span the full input range of the A/D converter. The amplification of the signals will increase the effective resolution of the digitized data.

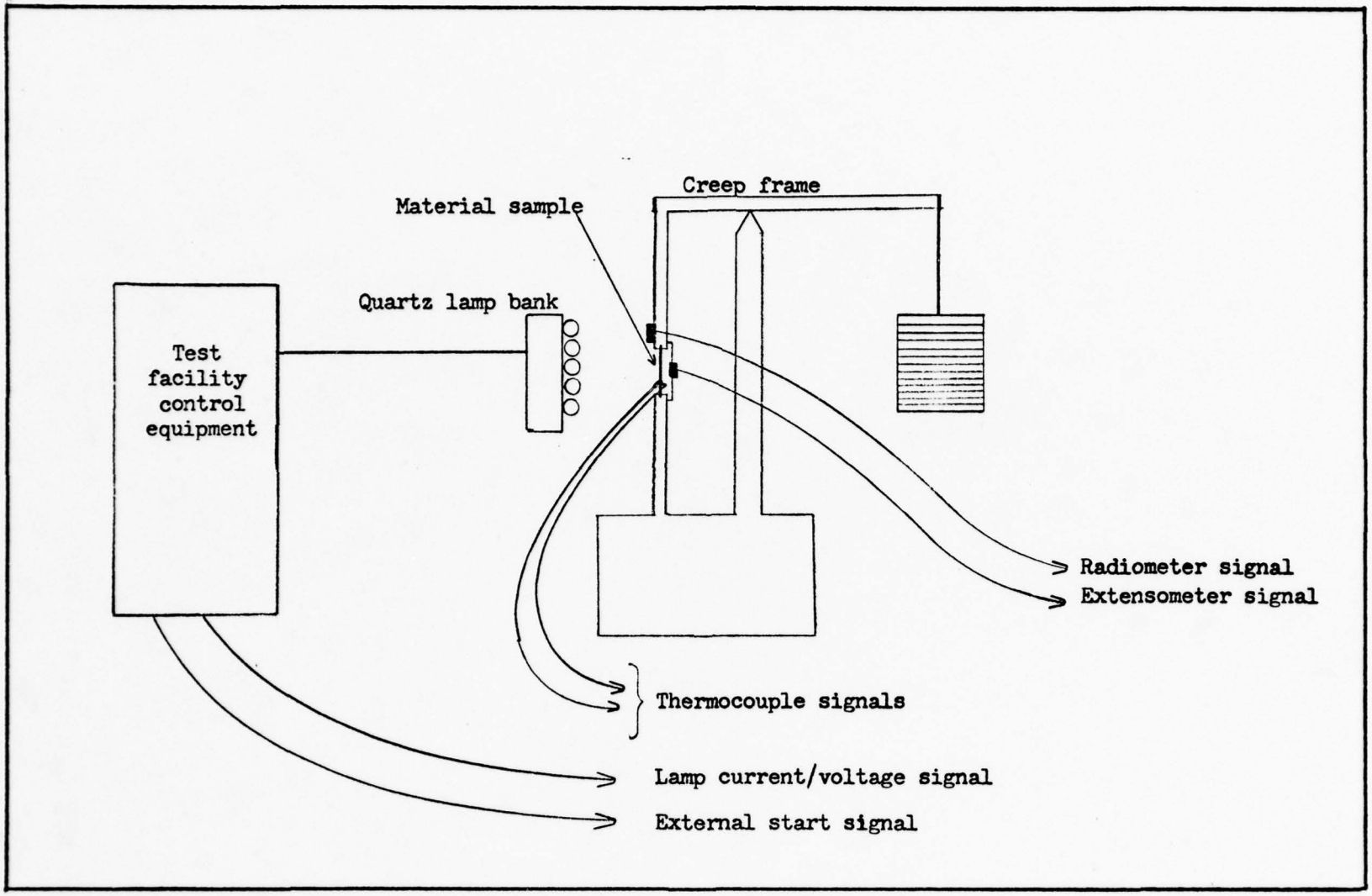


Figure 8. Data Acquisition System Signal Sources

To further increase the immunity of the signals to noise in the system environment, differential amplifiers are used in the data acquisition system.

Eight Ectron Model 687 DC amplifiers, whose specifications are listed in Ref. 4:1.1-1.5, are employed for the primary signal conditioning. These amplifiers have selectable gains of 10, 20, 50, 100, 200, 500, and 1000, with an additional vernier control, allowing a variety of input signal levels to be optimally matched with the A/D converter.

Temperature compensation of thermocouple reference junctions is another signal conditioning function performed. This eliminates the need for the $0^{o}C$ ice-water bath normally used at the reference junctions. Four of the Ectron amplifiers have Ectron Model 684 Ambient Temperature Compensators (ATC) built in. When used in conjunction with Ectron Model 683 Universal Thermocouple Adapters (UTA) at the thermocouple inputs, these amplifiers automatically perform temperature compensation of the thermocouple signals (Ref 4:9.1-9.4).

Since the system is configured to handle only positive signals, again due to A/D converter characteristics, transducers generating negative voltages must either have their signals inverted or must have a DC offset added. At the present time only the extensometer may occasionally have this problem, but adding an offset causes no difficulty since only changes in its output are of interest.

The signal conditioning amplifier outputs are also differential signals, and are transmitted to the inputs of the A/D converter. To connect the dissimilar plugs of the Ectron amplifiers and the A/D converter, a short cable is used. Corresponding pin assignments for the two connectors are given in Table A of Appendix A; additionally, the amplifier input

connections are listed in Table A2.

In reference to the external start signal discussed in the last section, when using a relay to couple to the data line it is usually not advisable to pass low-level signals through the relay prior to conditioning. Two reasons for this are that special handling is required for the thermocouple lines, and that the relay may introduce noise or interference. Instead, the relay connection can be made after the signals have been amplified and their voltage levels are greater, hence more resistant to noise.

Analog-to-Digital Converter

The analog-to-digital converter is a device which changes the continuous analog laboratory signals to a form which can be processed by a digital computer. It accomplishes this by sampling an electrical analog signal, and then generating a digital representation of the sampled signal. There are, however, two errors introduced by this conversion. The sampling error arises from sampling a continuous signal at a finite number of discrete intervals. But, by the sampling theorem, the signal may still be accurately reconstructed if the sampling rate is at least twice the highest frequency component of the original signal (Ref. 2:149). The other source of error is the quantization error caused by representing the signal amplitude by a finite number of possible levels. This error depends upon the size of each level, and for a binary representation there is an inherent error equal to one-half of the least significant bit (Ref. 12:115).

For the data acquisition system an ADAC Corporation Model 600-LSI-11 A/D converter is used. This device actually performs three distinct functions: multiplexing of the input data channels, sampling and holding the signal voltage level, and analog-to-digital conversion. Figure 9 is a

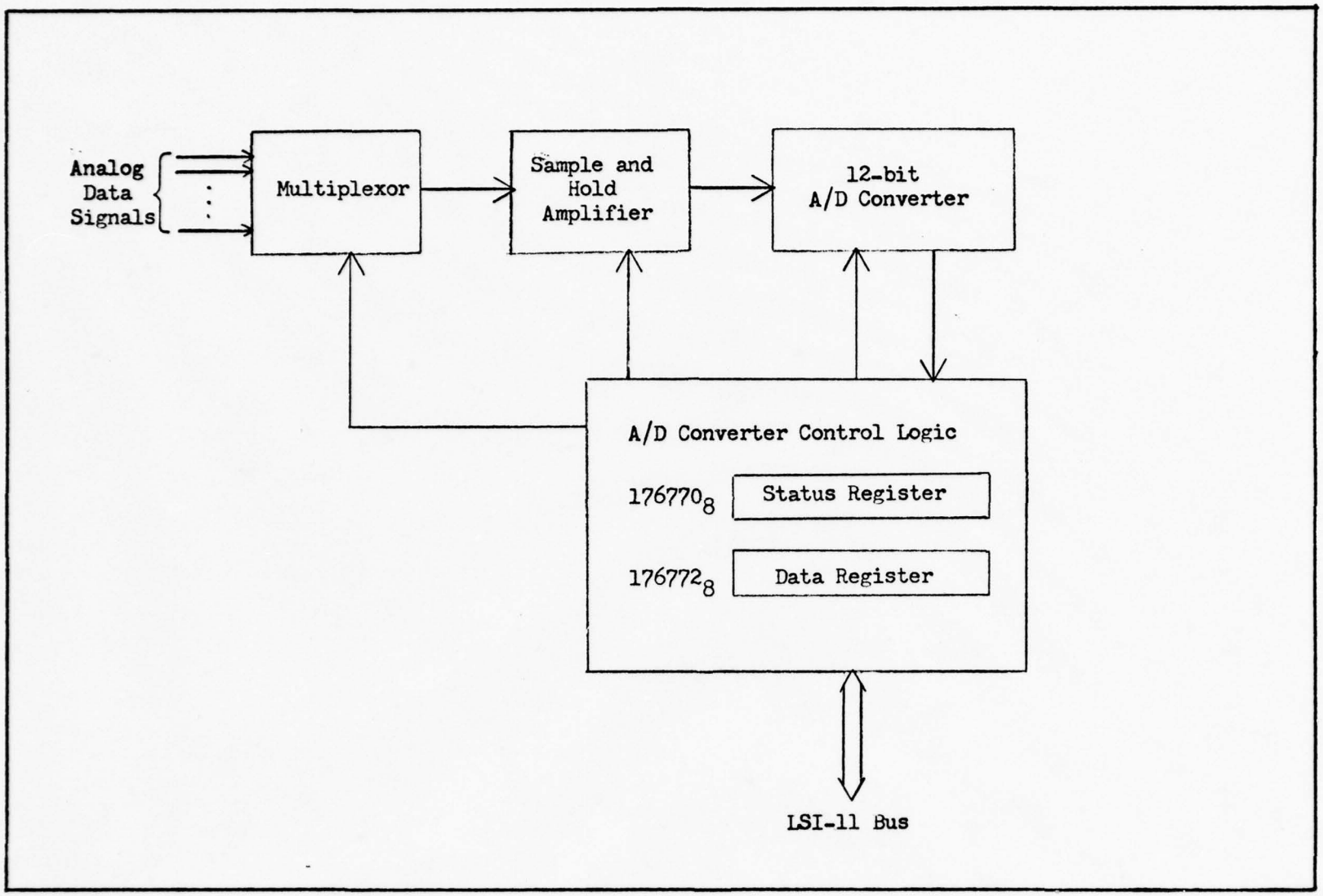


Figure 9. A/D Converter Block Diagram

block diagram of the major components of the A/D converter.

A multiplexor enables several differential signals to be sampled by selecting one of the eight input channels. The signal thus selected enters a sample and hold amplifier which samples the signal and maintains the voltage present long enough for A/D conversion. A complete conversion takes 15 microseconds, but by using the sample and hold amplifier which samples the signal over only 20 nanoseconds the effective aperture time during which the A/D converter is actively connected to the signal inputs is reduced (Ref. 3:13).

The A/D converter then digitizes the sampled analog signal using the technique of successive approximations. Briefly, this involves successively comparing the input signal to a reference voltage and then setting one of the bits in a digital word. A more detailed explanation of the A/D conversion technique is given in Reference 12. The A/D converter digitizes with a 12-bit resolution, corresponding to one part in 4096. Over the 0 - 10 V input range for which the converter is configured this means that the smallest distinguishable voltage is 2.44 mV, and that there is a quantization error of $\pm$ 1.22 mV. To achieve maximum resolution of an input signal it is desirable that the range of the signal cover as much as possible of the A/D converter input range. For example, if the input signal varies from 0 - 100 mV, it could be resolved to only about one part in 40 (40 x 2.44 mV $\simeq$ 100 mV). If the signal were amplified by 100 times, then the range of the signal input to the A/D converter becomes 0 - 10 V and the maximum possible resolution can be obtained. It is for this reason that the signal conditioning of low-level signals discussed in the last section includes amplification.

The ADAC 600-LSI-11 has a status register that allows the LSI-11 to

govern data sampling, and a data register that holds the digitized data. The computer bus addresses for these two registers are 176770_8 and 176772_8, respectively. The register bit designators are depicted in Figure 10 for those bits referenced by data acquisition system software. A sample is taken when a multiplexor address is loaded into the status register (bits 8-14) by the LSI-11. After allowing 5 microseconds settling time for the multiplexor and sample and hold amplifier, and 15 microseconds for conversion, the done status bit (bit 7) of the status register is set and the digitized data is available in bits 0 - 11 of the data register. From there the LSI-11 reads the data.

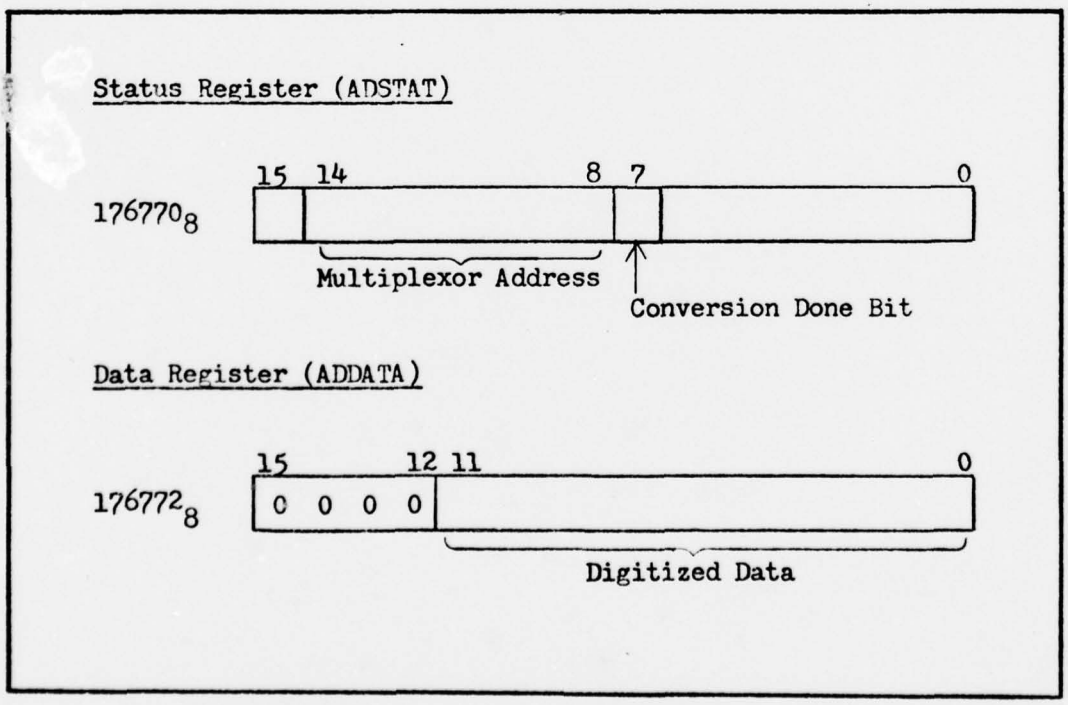


Figure 10. A/D Converter Status and Data Registers

Microcomputer and Associated Hardware

The central element of the data acquisition system is the LSI-11 microcomputer. Together with its related hardware and peripherals the LSI-11 controls the sampling of data by the A/D converter; stores, manipulates, and transmits data; and communicates with the user. The discussion of the elements comprising the LSI-11 computer system will be divided into the following sections: the processor, memory, system clock, teletype, modem, high speed tape reader, system backplane, and power supply. The information presented will only be that which is required for an understanding of the system configuration and the design of the LSI-11 software.

Processor. The Digital Equipment Corporation LSI-11 microcomputer is a 16-bit microcoded machine which executes the basic instruction set of the DEC PDP-11 family of minicomputers. It has eight general purpose registers and an extremely flexible set of addressing modes. Peripherals and up to 32K words of memory may be directly addressed over a bi-directional bus using sixteen multiplexed data/address lines and a number of control lines. Interrupts are vectored and priority is determined by the physical location of a device along a single daisy-chained interrupt line. A brief look at the LSI-11 system architecture is given in Ref. 13, and Ref. 6 is a very comprehensive system handbook.

The LSI-11 processor is available in a number of configurations; for this data acquisition system a KD11F module with 4K words of resident semiconductor RAM is used. In addition, a KEV-11 option is included which is a micro-ROM containing an extended arithmetic and floating-point instruction set for the processor. This option enables floating-point arithmetic instructions to be executed directly, rather than through software emulation, hence decreasing program storage requirements and increasing operation

speed. Table A3 contains a list of the module jumpers used to configure the processor.

There are two external signals required by the processor for proper operation, and which enter the system bus on two special control lines. One indicates to the processor that bus DC power is okay when high (BDCOK H), and functions as a processor reset when low. The other is a halt signal (BHALT L) which enables the processor to run when high, and halts it when low. The schematics for the switches used to produce these two signals are shown in Fig. 11. No debouncing is required for proper operation. Since the lines for these signals pass through the thermal flash control equipment they are shielded to protect against stray signals.

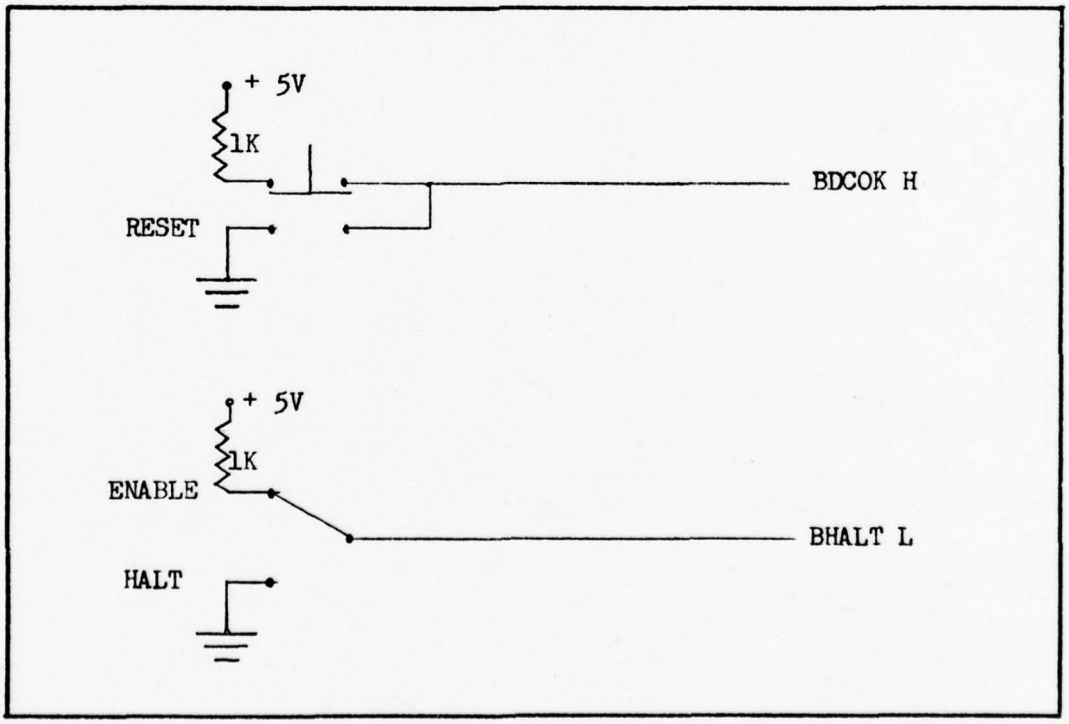


Figure 11. External Processor Control Signals

Memory. The data acquisition system was initially configured with 8K words of semiconductor RAM--4K on the processor module and another 4K on a DEC MSV11-B memory module. Later, when increasing program size necessitated it, an additional 8K words of RAM on two more memory modules were installed. A list of the jumpers selected on the three MSV11-B modules to control memory operation is in Table A4.

Clock. An external clock is used to provide timing for the data acquisition system. Fabricated by AFML/DOC and supplied for use with this system, it employs a one megahertz crystal controlled oscillator and some additional logic to produce TTL level pulses at one kilohertz. These pulses enter the LSI-11 bus through a special clock interrupt line (BEVNT) and cause the processor to trap to the address in memory location 100_8. The kilohertz rate is more than sufficient for the 0.01 second time base resolution required. The clock can be disabled by an externally mounted on/off switch.

Teletype. To communicate with the operator and load system programs a Teletype Corporation ASR 33 teletype is used. It includes a printer, keyboard, and low-speed paper tape reader, and operates at 110 baud. A DEC modification for the teletype, LT33, is used to allow the tape reader to be remotely controlled by the LSI-11 in order to load tapes. The teletype interface is a DEC DLV11 serial interface module that is configured for 20 mA current loop operation at 110 baud. The exact module options jumpered are given in Table A5. Device control/status and data registers in the interface are accessed by the LSI-11 to control teletype input and output (I/O). They will not be described here since the registers are the same in layout and operation as for the modem interface discussed next, with the exception of a device address of 177560_8 and an interrupt vector

of 60_8.

Modem. An Anderson-Jacobson A242 acoustic coupler modem is used by the LSI-11 to communicate with other computers over a telephone line. This device uses frequency shift keying modulation for asynchronous communication at rates for up to 450 baud (Ref. 1). For this system the modem is operated at 300 baud for compatibility with the SEL 86 and CDC 6600 modems. Interfacing with the LSI-11 is accomplished using another DLV11 serial interface module, configured for EIA RS232C operation at 300 baud. A device address of 175610_8 and interrupt vector of 300_8 were selected for the interface in accordance with the DEC convention for assigning DLV11 addresses (Ref. 6:5-26 to 5-27). The interface module jumper configuration is listed in Table A6, and the pin assignments for a cable constructed to connect the acoustic coupler to its interface are given in Table A7.

The serial interface contains four registers that are addressed by the LSI-11 to control the transmission and reception of data through the modem. The registers are depicted in Fig. 12 together with their bit designations. To transmit data a one-byte character is loaded into the low byte (bits 0-7) of the transmitter data buffer (MXBUF--address 175616_8) by the LSI-11. When the data has been transmitted, the transmitter ready bit (bit 7) of the transmitter control/status register (MXCSR--address 175614_8) is set, and an interrupt is generated if the transmitter interrupt enable bit (bit 6) has been set. Character data received from the modem is placed in the low byte (bits 0-7) of the receiver data buffer (MRBUF--address 175612_8), where it can be read by the LSI-11. The receiver done bit (bit 7) is set in the receiver control/status register (MRCSR--address 175610_8) and, as before, an interrupt is generated if the receiver interrupt enable bit (bit 6) has been set. The MRCSR also has a data set status bit

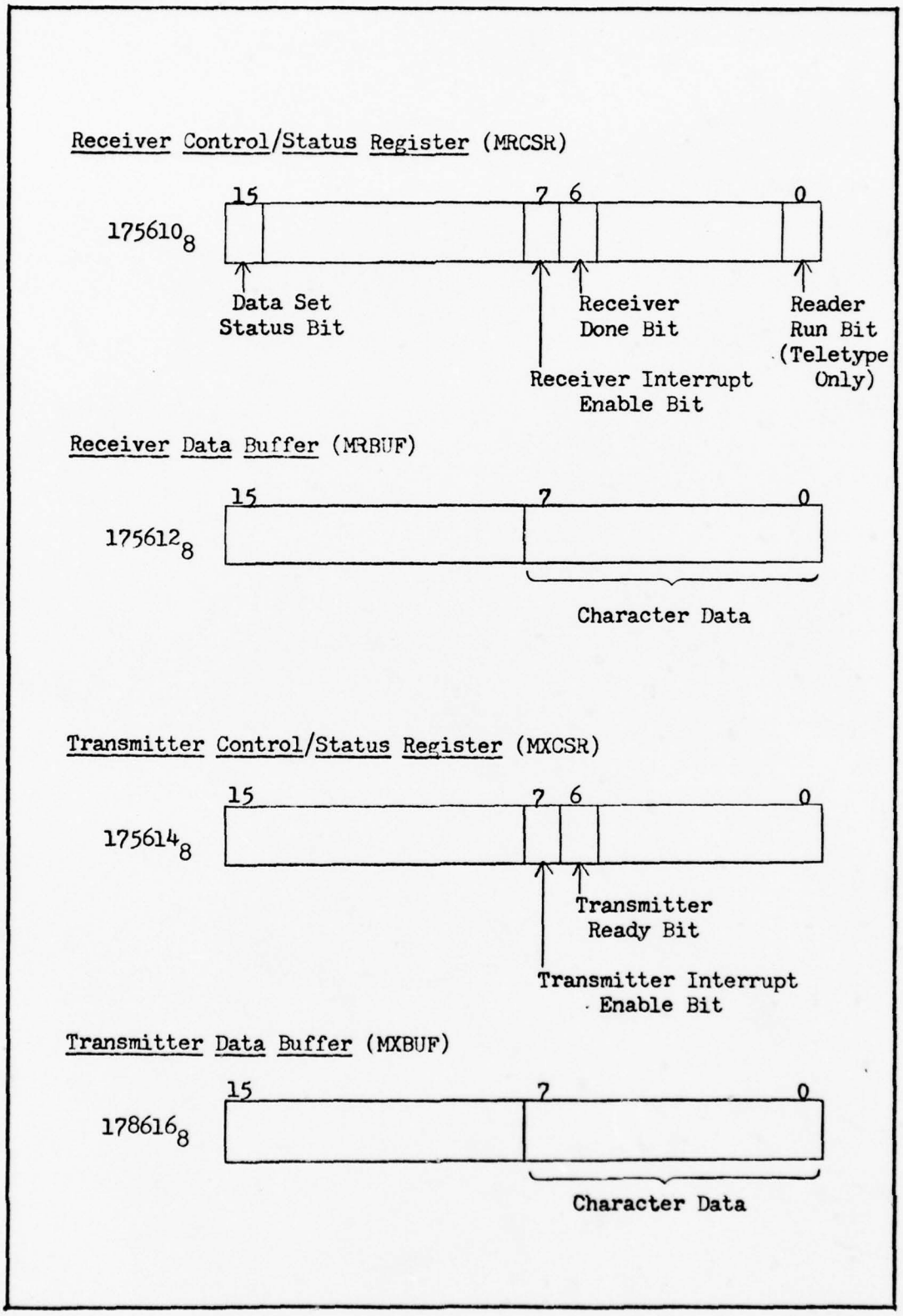


Figure 12. Modem Serial Interface Registers

(bit 15) that is set when the carrier detect or clear-to-send, and the data-set-ready signals of the EIA modem interface are asserted. This indicates that the acoustic coupler is ready for data transmission or reception.

High-Speed Paper Tape Reader. To facilitate program development a high-speed paper tape reader was connected to the data acquisition system. It is described here since the absolute loader developed for it is discussed in Chapter V. The reader used is a Control Logic Model LHR punched paper tape reader. A DEC DRV11 parallel interface module is used to interface the reader to the LSI-11. Its register bit designations are shown in Fig. 13. A character read is initiated when the step reader bit (bit 0) of the output data buffer (PROUT--address 167772_8) transitions from one to zero. After the character is read it is placed in the low byte (bits 0-7) of the input data buffer (PRIN--address 167770_8) and the not-ready bit (bit 8) is cleared.

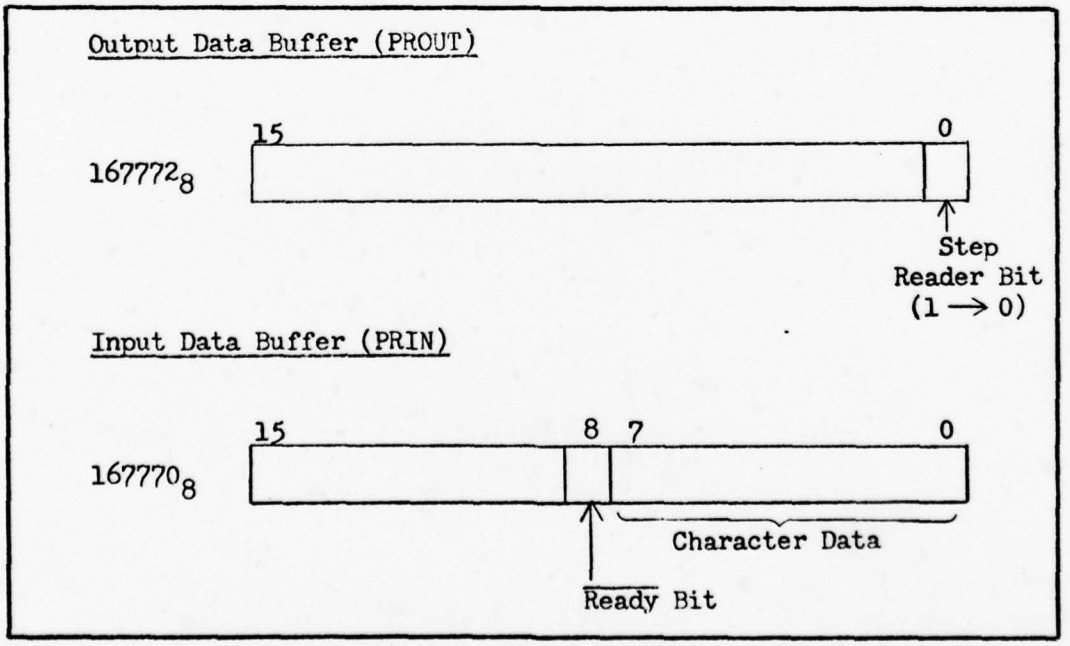


Figure 13. High-Speed Paper Tape Reader Parallel Interface Registers

System Backplane. All of the interface and memory modules, as well as the LSI-11 processor and the A/D converter, are inserted into the system backplane. Figure 14 illustrates the configuration of the system backplane and the placement of modules. The system backplane is made up of two DEC H9270 backplane and card cage assemblies which contain the modules, and distribute power and LSI-11 bus signals to each of them. The two backplanes are connected using DEC M9400YD and M9401 modules with two BC05L-XX cables to transfer bus signals between the backplanes. A TEV11 120 Ohm module is also used to maintain proper bus termination. The system backplane, containing the computer modules, is mounted in a shielded area of the test facility control console.

Power Supply. The power supply for data acquisition system produces +5 V and +12 V for all modules plugged into the system backplane, and for the system clock. A Standard Power Model SPS 250D dual power supply, rated at 8 A @ + 12 V and 12 A @ + 5 V, is used.

Summary

This chapter has described the hardware components of the data acquisition system, and presented some system design considerations. The hardware organization, introduced in Chapter III, has been examined in greater detail. Chapter V will expound on the data acquisition system software.

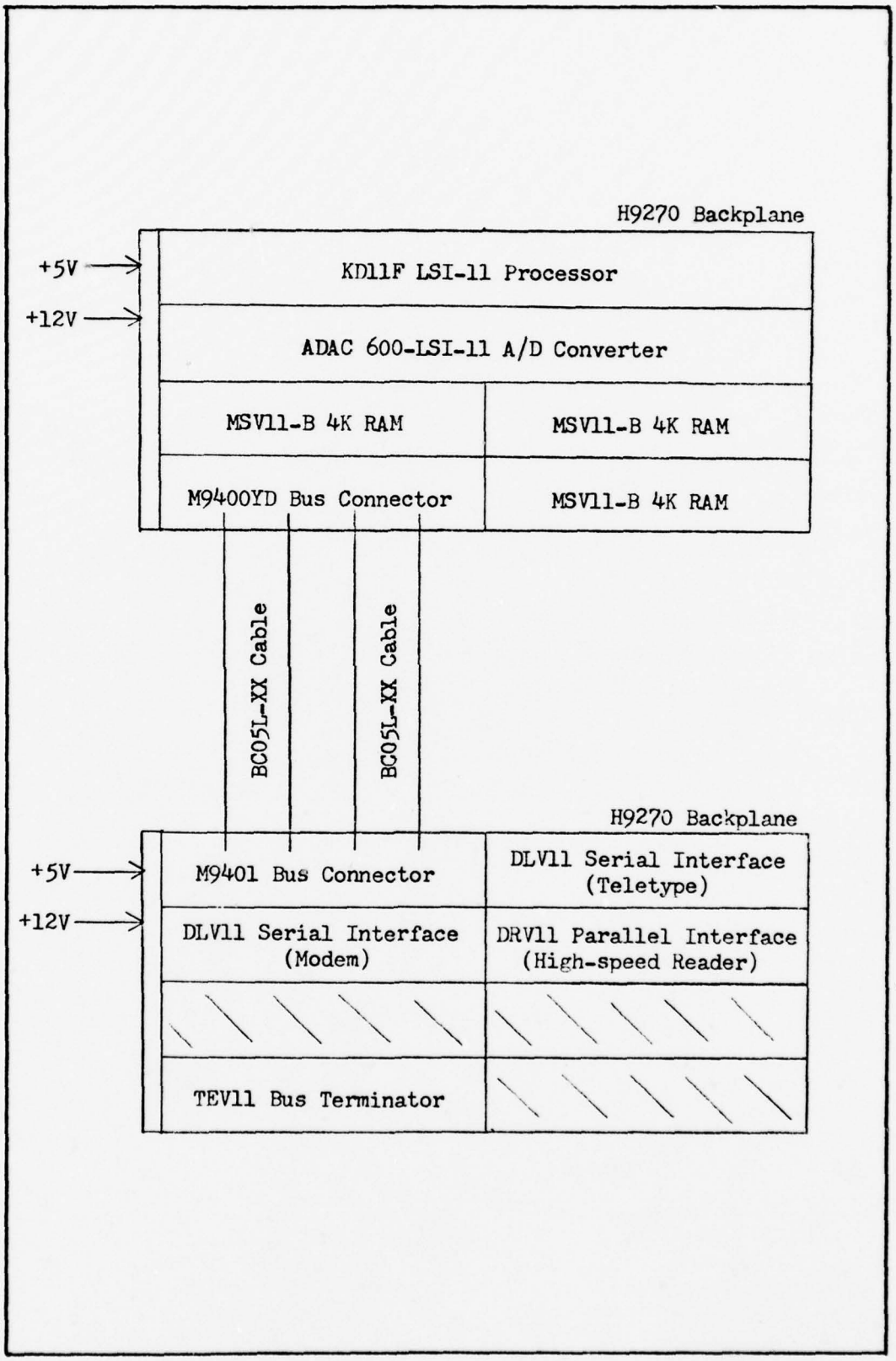


Figure 14. LSI-11 System Backplane and Module Configuration

V. Data Acquisition System Software

Introduction

The objective of this chapter is to describe the software employed in the data acquisition system, including its design development and implementation. Overall software organization was described in Chapter III; specific algorithms and software design considerations will now be analyzed. The discussion of system software will begin with an examination of the general development approach. This is followed by a description of the FORTRAN algorithms which perform the major system functions, and the assembly language routines which support system operation. There will also be a brief mention of other software, not directly a part of the data acquisition system, which was written in the course of this investigation for system development and demonstration. No detailed examination will be made of the realization of any specific algorithms since the source code contains detailed comments which document the software. Only the basic structure of the algorithms and the functions performed by various routines are discussed, and flow charts are used only to clarify system operation or to point out certain characteristics.

Software Development

The software for the data acquisition system has been developed using a modular approach in order to fulfill the software constraint, discussed in Chapter II, that the system programs be easily modified and maintained. The system software has also been designed to be general enough to handle changing system requirements, again to satisfy a software constraint.

In designing the software for the data acquisition system, each of

the functional system requirements was identified and a modular section of code was developed to handle that function. The functional requirements were defined in Chapter II as the data acquisition task, management of system parameters, transmission of data, limited on-line data reduction, and time-sharing terminal emulation. The routines which perform these functions are written in FORTRAN because of the additional software constraint that a higher-order-language be used to facilitate program modifications by the user. The specific subroutines used are discussed later in this chapter. Rather than linearly chaining the functional modules together, it was decided to operate them under control of a command module to give the user the capability to perform functions only as required. A central module was also developed to control the acquisition, peak calculation, and transmission of data so that these functions may be done automatically if needed as demanded in the system specifications. Some aspects of system operation required the development of subroutines, realized in assembly language, to perform particular functions in support of the main routines. These functions include sampling the A/D converter, providing a time base and a means of measuring sampling intervals, and performing modem I/O. The assembly language routines will also be discussed individually later in this chapter.

During the development of the software for this system many other decisions have been made concerning tradeoffs between structure, efficiency, and ease of operation. A modular software structure has been used throughout, as stated above; however, in an effort to reduce memory overhead to keep the program within the 8K words initially available, more than one function is performed in some subprograms. While this decreases the

overall level of modularity, combining only those functions pertaining to a single area of system operation maintains software integrity. As an example, the test parameter handler is used to both change and print the system test parameters, but performs no other functions. Memory and execution efficiency are sacrificed at other times, however, to increase the ease of system operation for the user. For instance, input is checked for validity when possible, and error messages are provided. The user is also prompted whenever input or some action is required. Insuring that the system is kept general requires additional tradeoffs along these lines. Further design considerations are discussed as they arise in the examination of system software.

System Software (FORTRAN)

The data acquisition system uses FORTRAN code to perform the major system functions, as discussed in the last section. The FORTRAN routines were briefly described in the software organization section of Chapter III (also see Fig. 5) and will now be examined more closely following some comments concerning the use of FORTRAN in the data acquisition system.

FORTRAN Usage. The development system, briefly described under the assumptions of Chapter II, used to write software for the data acquisition system operates under a DEC Real-Time 11 (RT-11) operating system (Ref. 10). The FORTRAN object code produced by the compiler on this system requires the RT-11 monitor to do certain functions, including console terminal I/O. Since the data acquisition system does not have a disk, it will not support the normal RT-11 operating system. Therefore, an RT-11 simulator which is supplied as a part of the development system's FORTRAN library is linked with the FORTRAN code to provide the required functions

for a stand-alone system such as the data acquisition system being discussed (Ref. 9:1-17 to 1-18). With the RT-11 simulator the normal formatted FORTRAN I/O statements can be used to communicate with the teletype. Additionally, special terminal I/O operations performed by some terminal keys are maintained. These include end-of-file (CONTROL Z), rubout (RUBOUT), and halt/restart terminal output (CONTROL S, Q, and O); they are described fully in Ref. 10:2-12 to 2-13.

Free-form input is used whenever possible to simplify operation of the system by the user. To achieve this, character strings are input as variable length with a carriage return delimiting the end of the string. Floating-point numbers are input using F10.0 format, and integers use I10 since the FORTRAN input statements allow decimal points, commas, and carriage returns to override the declared fields and to delimit numeric data. The version of FORTRAN used also has a format specifier (Q) which is used to find the number of characters in any input line (Ref. 8:6-11). Using this capability a program can determine whether a zero has been input, or only a carriage return. By making this distinction, the system routines which input data can allow the operator to only enter a carriage return if he does not wish to change a current value, or to enter a zero if that is the data to be input. This is not done if no defaults are to be permitted for a variable. Detection of a carriage return input is also used to exit from a series of inputs in some instances. Whenever input is requested from the operator, typing a CONTROL Z generates an end-of-file error which is trapped by the input statement. In all cases this is used to force a return to the command input routine, and allows the technician to abort an operation.

Overview. The discussion of FORTRAN routines which follows examines: the command input decoder, the test and plot parameter handlers, the acquisition control subroutine, the data acquisition subroutine, the peak data calculation subroutine, the data transmission subroutine and its supporting FORTRAN routines, the time-sharing terminal emulation routine, and the BLOCK DATA section. References are made at times to functions performed by the assembly language routines; these will be discussed later in this chapter.

Command Input Decoder (CMND). The purpose of the command input decoder is to accept commands entered by the technician and call the appropriate FORTRAN subroutines to perform the requested operation. The strategy behind this is to allow the user flexibility in determining which system operations are to be performed at any time. It also enables other FORTRAN modules to be added, deleted, or modified as the need arises, without affecting the overall system structure. The basic algorithm for CMND is depicted in Fig. 15.

Since CMND is the first routine entered in the data acquisition system, it is used to do any necessary system initializations. At the present time the only one required is to enable modem receiver interrupts, so that if the technician should call up another computer before entering the data transmission or terminal emulation routines no messages will be lost. A heading is printed to identify the program, and a period (.) is used to prompt the operator to type in a command.

The command is input as a character string and is divided into a command and argument, where the argument is used to further specify the operation indicated by the command. System commands and arguments are

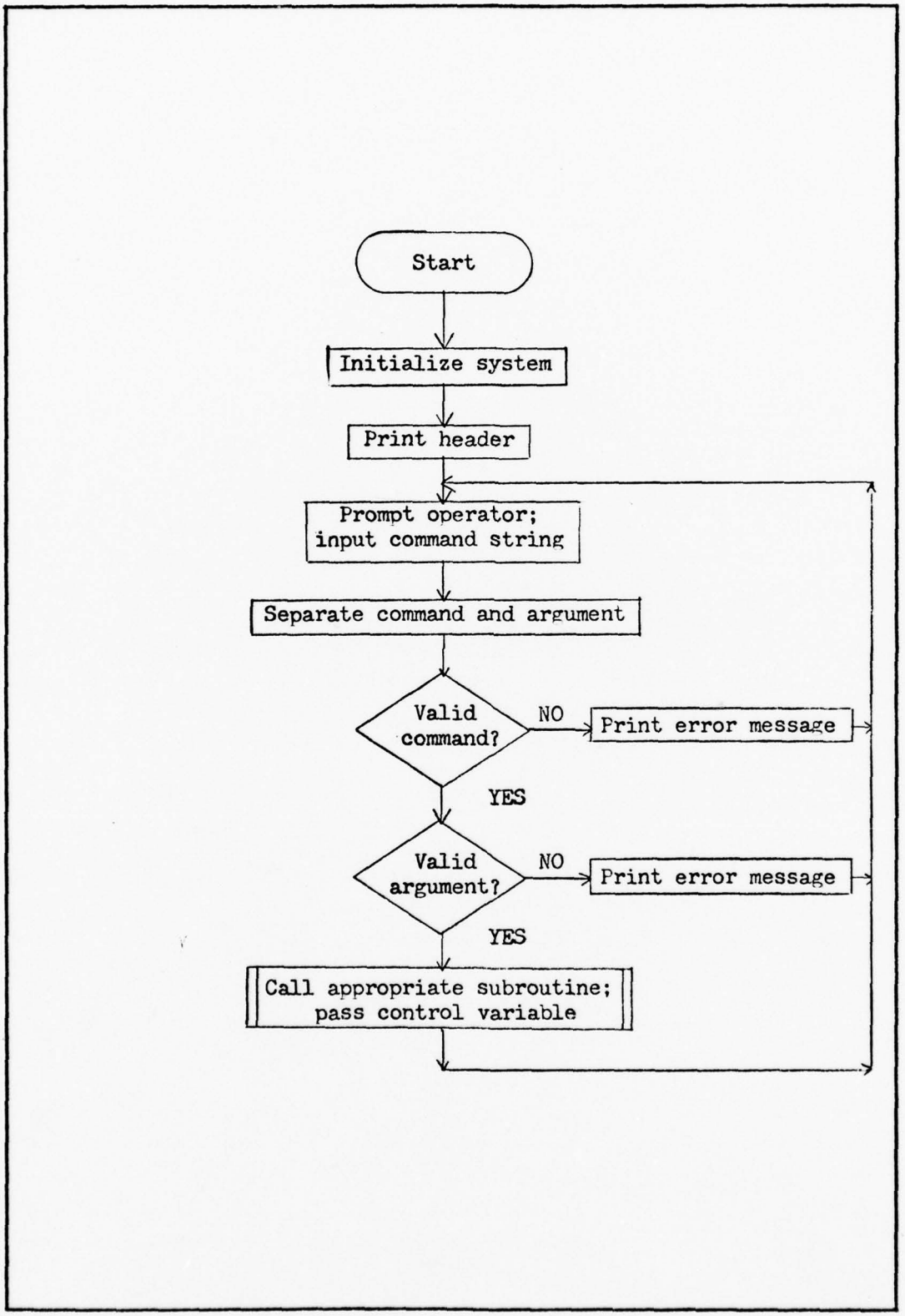


Figure 15. Command Decoder Algorithm

summarized in Table II, along with a list of the functions they control to satisfy system requirements. Both the command and argument are treated as variable length character strings that are manipulated by special subroutines available in the FORTRAN library (Ref. 10: Appendix O). This permits flexibility in assigning commands for the system since no fixed length is required. The commands are presently chosen to be concise and descriptive, but may be easily modified.

A command is decoded by matching it against valid commands, and when one is found, checking the argument with legal ones. An error message is printed if the command or argument cannot be decoded. Otherwise, the appropriate subroutine is called to do the indicated action. For the routines that perform more than one function a control variable is passed which indicates which operation is to be performed.

Test Parameter Handler (TEST). The test parameter handling subroutine manages the test parameters, discussed in Chapter II, that govern the acquisition of data by the system. Figure 16 shows the flow chart for this routine which performs two distinct functions: 1) change test parameter values and 2) print values currently in effect. As stated in the last section, when called from CMND a control variable is passed to TEST to determine which function is done.

1. Change Test Parameters. The data acquisition program is loaded with default values set for the test parameters, among others, by a BLOCK DATA section which was mentioned in Chapter III and will be described later. To change the values of the test parameters for varying sample test requirements, this routine is used. A set of parameters for each channel specified by the operator are input, and checked against

Table II

Data Acquisition System Commands

Command/Argument	Function (Routine(s) Called)
CHNG TEST	Change test parameters (TEST)
CHNG PLOT	Change plot parameters (PLOT)
CHNG	Equivalent to 'CHNG TEST' followed by 'CHNG PLOT' (TEST, PLOT)
PRNT TEST	Print test parameters (TEST)
PRNT PLOT	Print plot parameters (PLOT)
PRNT	Equivalent to 'PRNT TEST' followed by 'PRNT PLOT' (TEST, PLOT)
ACQR	Initiate data acquisition test run (ACQR)
ACQR ON	Enable automatic re-acquisition (ACQR)
ACQR OFF	Disable automatic re-acquisition (ACQR)
PEAK	Print peak data information
PEAK ON	Enable automatic printing of peak information
PEAK OFF	Disable automatic printing of peak information
TRAN	Transmit data (TRAN)
TRAN ON	Enable automatic transmission of data (TRAN)
TRAN OFF	Disable automatic transmission of data (TRAN)
TTY	Transparently emulate time-sharing terminal (TTY)

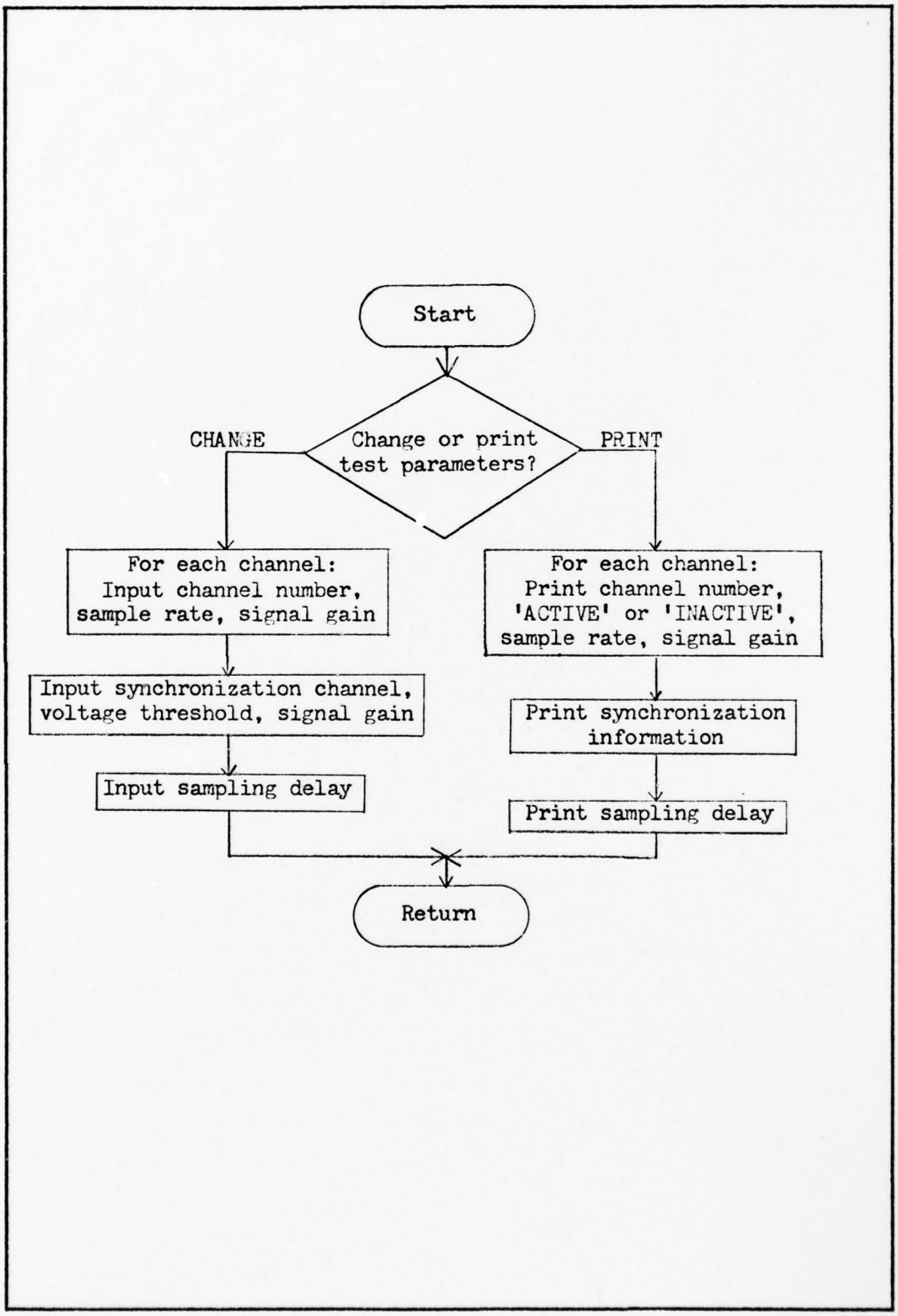


Figure 16. Test Parameter Handler Algorithm

allowable values. Any errors are brought to the attention of the operator. If only a carriage return is entered when a channel number is requested, the acquisition start synchronization parameters are input. A zero input for the synchronization channel indicates acquisition is to be started manually; any other valid channel number indicates that channel is to be used for synchronization purposes. Then the sampling delay is input and checked.

As discussed earlier, a carriage return entered as input maintains the value currently in effect, except for a channel number request where it indicates an end of channel parameter inputs. A CONTROL Z forces a return to CMND.

2. Print Test Parameters. When a request is made to print test parameters, the values currently in effect are printed on the teletype. For each channel the channel number is printed, then 'ACTIVE' or 'INACTIVE' depending if any data is to be sampled, the sampling rate, and the signal gain set on the amplifiers. The synchronization information is printed, and whether acquisition start is automatic or manual. This is followed by the sampling delay.

Plot Parameter Handler (PLOT). The plot parameter handling subroutine manages the system plot parameters passed to the user's CDC 6600 plotting programs. They were described in Chapter II. Like TEST, PLOT performs two functions controlled by a variable passed from CMND.

1. Change Plot Parameters. The system plot parameters are also set as defaults when the data acquisition system program is loaded. To change them, when this routine is entered a channel number is input. If just a carriage return is typed the routine exits, otherwise the plot

code for that channel is input and checked. It must be an integer between zero and 9999. If the plot code is 9999, then a set of 20 special codes is input.

The special codes are logically arranged in a two-dimensional array of eight rows corresponding to the system channels, and twenty columns for the codes. Since arrays are declared as one dimensional to increase flexibility, as discussed in Chapter III, an arithmetic function statement is used to perform the correct mapping based on data in a COMMON block. Other routines which access the special codes also use a mapping function.

2. Print Plot Parameters. To display the present values in effect for the plot parameters the print plot parameters routine is used. For each channnel the channel number, plot codes, and special codes, if any, are printed.

Acquisition Control Subroutine (ACQR). The purpose of the acquisition control subroutine is to govern the data acquisition for a test run, and to permit automatic peak data information display, automatic transmission of data, and automatic re-acquisition. By allowing ACQR to call the peak data routine and the data transmission routine, the operations they perform may be done automatically after each test run without requiring the technician to initiate them through CMND. This increases turn-around time for the system, and eases system operation for the user. For these same reasons ACQR may automatically re-initiate data acquisition for another test.

The ACQR subroutine performs three functions, determined by the control variable passed from the command input decoder.

1. Acquisition Control. The acquisition control algorithm, pictured in Fig. 17, handles the data acquisition control and automatic operations discussed above. The routine inputs the acquisition run parameters--run number, title, and run time--which are the final set of parameters used to control data acquisition and the treatment of data in the network. No defaults are permitted since the run number must be unique, and with the title and run time it must be typed in so there is always a printed copy for record purposes.

On the basis of the run time, sampling delay, and sampling rates for the system the number of samples which will be taken during the run is calculated. The total may not exceed the available data array space, or an error message is printed. A call is made to the data acquisition routine which does the start synchronization and acquisition of data for the run. If no data was taken because the user aborted, the routine returns to CMND. Otherwise, a call is made to the peak data routine which automatically prints peak data information for the run if that function has been enabled. Likewise, the data transmission routine is called which transmits the data for the run if that is to be done automatically. If automatic re-acquisition has been enabled, determined by checking a flag, then another set of acquisition run parameters is immediately requested. If not, the routine returns to CMND. As before, a return to CMND is forced if a CONTROL Z is ever entered as input.

2. Enable Automatic Re-acquisition. To enable automatic re-acquisition a flag is set which is checked during the acquisition control sequence, above.

3. Disable Automatic Re-acquisition. The automatic re-acquisition

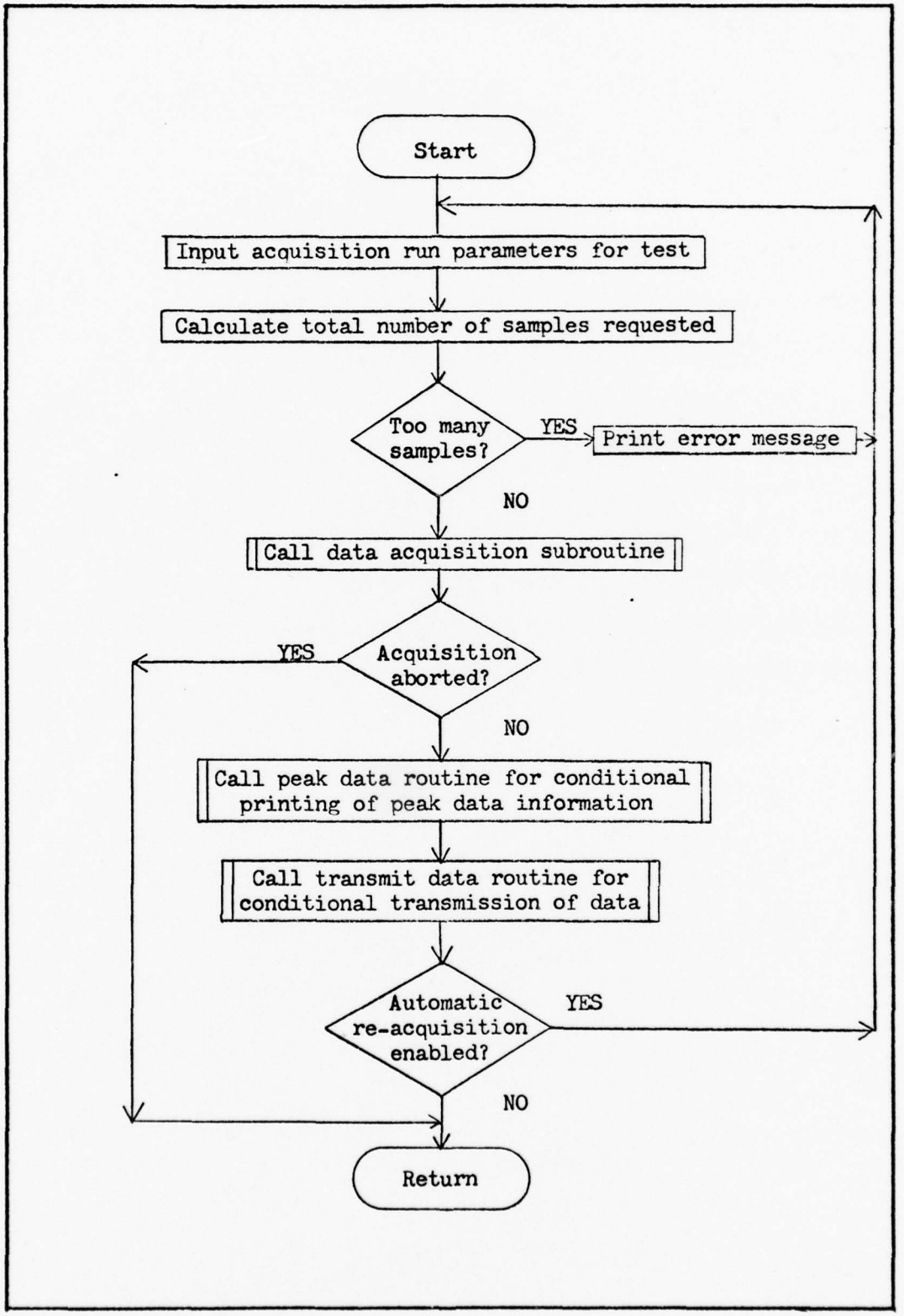


Figure 17. Acquisition Control Algorithm

flag is cleared to disable that function.

Data Acquisition Subroutine (ACQDAT). The data acquisition subroutine, called only from ACQR, synchronizes the start of data acquisition and collects data for the run. The algorithm is shown in Fig. 18. Variables and constants to be used later are initialized first, so few computations will be required during acquisition. Then the system clock is turned off and set to zero. The software timers for each channel are loaded with the sampling interval, which is calculated from the required sampling rates for the channels. Timers for inactive channels are disabled. A call is then made to the A/D converter sampling subroutine (ADC) to force a single initial conversion. This is required since data in the A/D converter data buffer is undefined for the first conversion after power-up, and must be cleared.

If manual synchronization has been specified in the synchronization test parameters, the technician is prompted and the program waits until a carriage return is typed. If data acquisition is to be automatically synchronized with an external start signal, ADC is continually called to sample the synchronization channel until the signal crosses the synchronization voltage threshold. For either manual or automatic synchronization a CONTROL Z aborts the data acquisition run and an error return is made.

The system clock is turned on when data acquisition starts, and the teletype bell is rung to signal the operator. A set of initial data is then taken on all actively sampled channels, and stored in a compacted format. The sample time is also stored. A test is made to see if the total test time, calculated from the run time and sampling delay, is up.

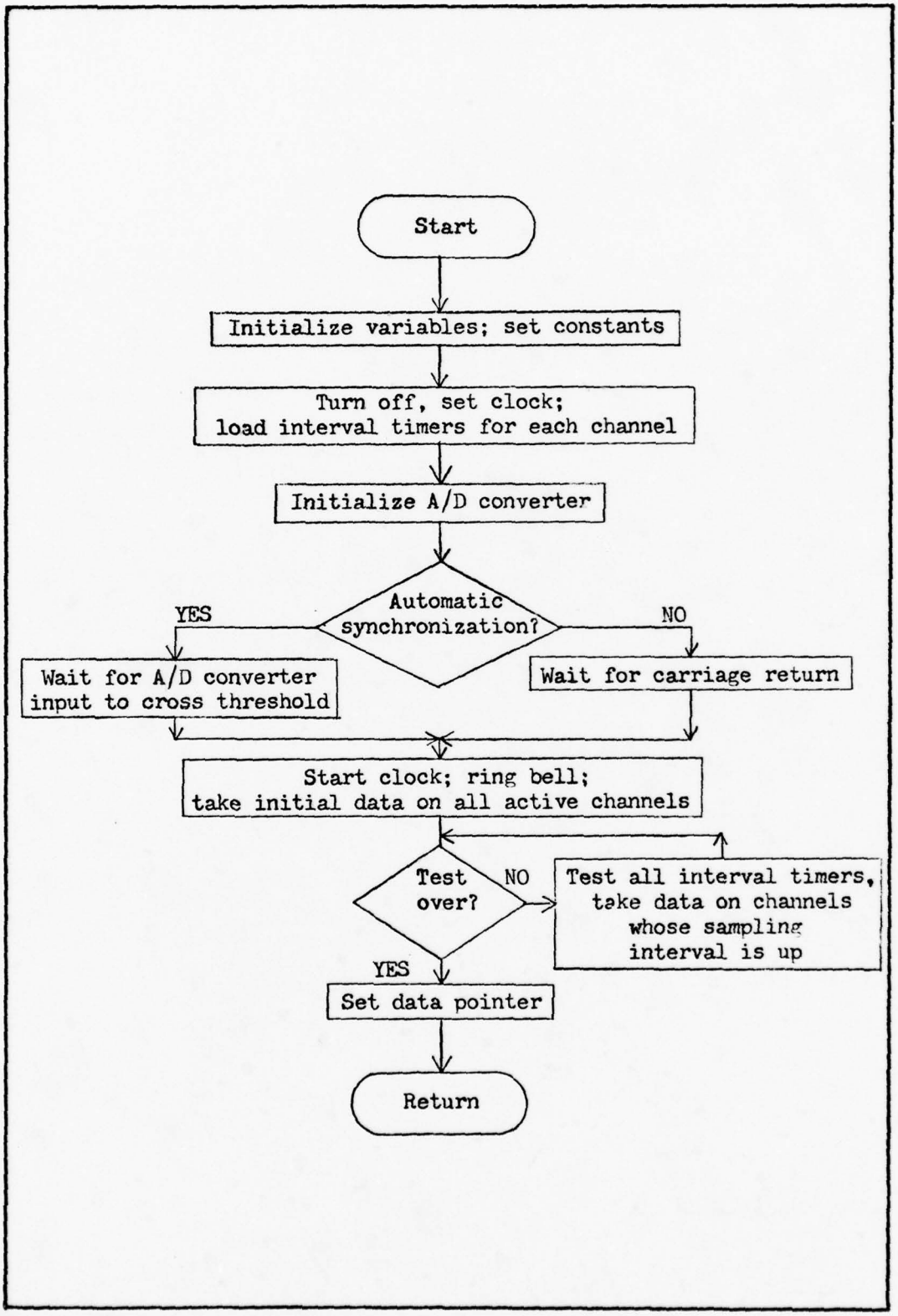


Figure 18. Data Acquisition Algorithm

If the test is not over the timers for each channel are scanned, and for those which indicate an expired sampling interval, data is taken on that channel. After all the channel interval timers have been tested once, the test time is again checked. Sampling continues until the test is over. At the end of the test a pointer is set to indicate the end of the test run data in the data array.

Peak Data Subroutine (PEAK). The peak data subroutine is used to perform four functions: 1) calculate and print the peak data information for the latest run, 2) conditionally print peak information, 3) enable automatic printing of peak information, and 4) disable automatic printing of peak information. The function is controlled, as before, by a variable passed from the calling routine, which may be CMND or ACQR.

1. Print Peak Data Information. To print peak data information for a channel, a search is made of the compact data array for the peak data by unpacking the channel and data at each point. As the entire array is searched for the data of a particular channel, the peak data and the time at which it was taken are found. The millivolt data is then calculated from the 12-bit form generated by the A/D converter by multiplying by a 2.44 mV/bit conversion constant. The resulting number is divided by the signal gain set for the channel so that true signal magnitude is reflected. Likewise, the time at which the peak data was taken must be divided by the clock frequency to find the time in seconds. The channel number, peak data in mV, and time in seconds are printed. This sequence is repeated for all active channels.

2. Conditionally Print Peak Information. When a call is made from ACQR to automatically print peak data information a flag is checked,

and if set, the algorithm just discussed is executed to print the peak information. If the flag is not set the routine returns.

3. Enable Automatic Peak Information Printing. The print peak flag checked above is set to enable automatic printing of peak data information.

4. Disable Automatic Peak Information Printing. The automatic print peak flag is cleared to disable that function.

Data Transmission Subroutine (TRAN). Transmission of test data and parameters to the SEL 86 computer system for storage and re-transmission to the CDC 6600 is handled by TRAN. It performs four functions: 1) transmit data from the last test run, 2) conditionally transmit data, 3) enable automatic data transmission, and 4) disable automatic data transmission.

1. Transmit Data. Communication with the SEL 86 is through that system's Terminal Support Subsystem (TSS), described in Ref. 15 . Other approaches were attempted, but the SEL 86 Real Time Monitor operating system does not presently provide another method of communication with low-speed data lines. Hence, existing TSS software must be used. It had been desired to use a FORTRAN program running on that system to handle data communication with the LSI-11 data acquisition system. Instead, data can only be transmitted to the SEL 86 for storage using the TSS editor. This requires that the data acquisition system appear to TSS to be a user typing in data at a time-sharing terminal. The basic algorithm for communicating with the SEL 86 is depicted as the flow chart of Fig. 19. Communication is facilitated using three other FORTRAN subroutines discussed in succeeding sections of this chapter.

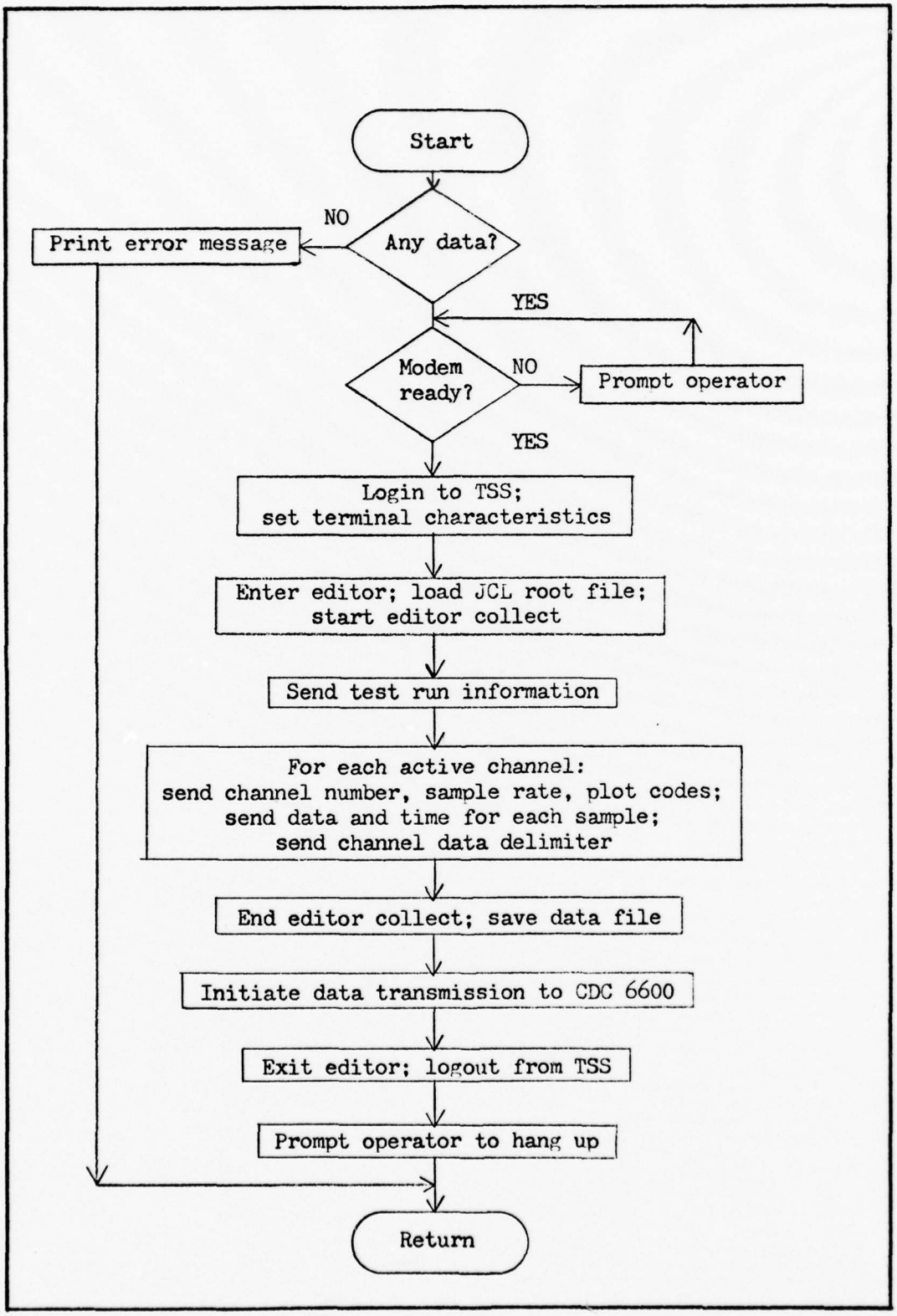


Figure 19. Data Transmission Algorithm

When a request is made to transmit data to the SEL 86, the data array is checked by TRAN to see if any data is available. If there is none, an error message is printed and the routine returns. If there is data, the modem is tested to determine if it is ready, and a prompting message is displayed to the operator if not. When the modem is ready TRAN sends a login message to TSS, and sets TSS terminal characteristics for further transmission. The TSS editor is entered and a file is loaded containing the CDC 6600 JCL necessary to run the user's plotting routines. A 'collect' command is sent to the editor so the data which follows is appended to the root file just loaded.

The format of the data stored on the SEL 86, based on discussions with the user, is pictured in Fig. 20. All data is transmitted as character strings; numeric data is changed to character strings for transmission using ENCODE statements. The run number, title, run time, and total sampling time are transmitted first. Then for each active channel the following test and plot parameters are sent: channel number, sampling rate, plot code, and special codes, if any. The data in millivolts and the sample time in seconds are then transmitted for each data point. They are calculated from the raw data as discussed for the peak data routine. After the data for the channel a data list delimiter is sent to indicate the end of data for that channel. This is repeated for the other active channels.

After all data and parameters have been sent, the TSS editor 'collect' is terminated. The JCL and data is next stored on a permanent SEL 86 system disk file, using the run number as the file name, by executing the TSS 'save' command. If another file exists under the same name it is scratched, hence unique run numbers are required. However, this allows

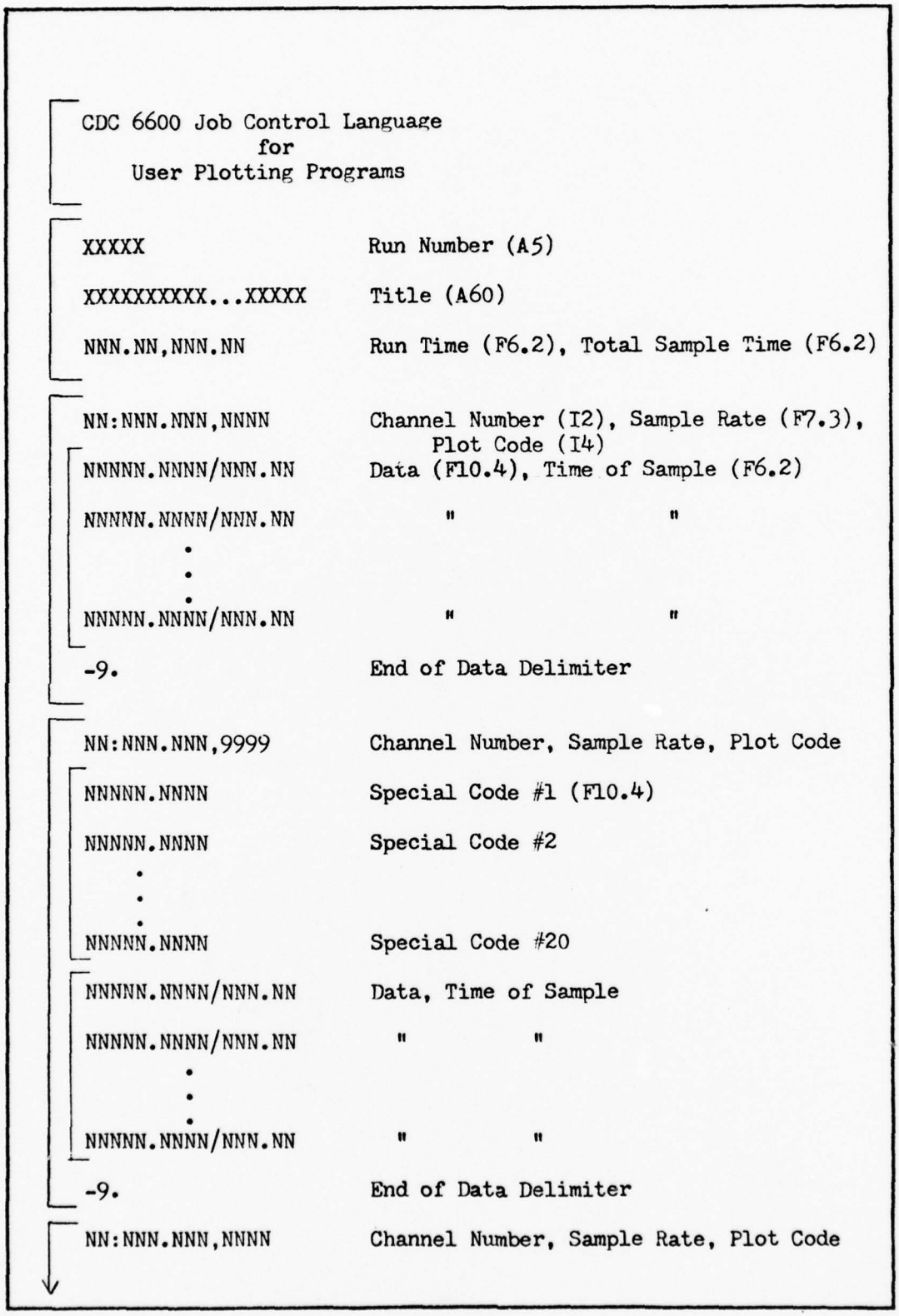


Figure 20. Transmitted Data File Format

a set of data to be superceded if necessary.

To initiate retransmission of data from the SEL 86 to the CDC 6600 a 'run' command is sent to TSS, causing the file just created to be transmitted by the SEL 86 to the CDC 6600 using a high speed data link. The JCL and data in the file appear to the CDC 6600 to be a normal batch job, and the user's plotting routines are subsequently executed using the test data for input. While this is occurring, TRAN logs out of the editor and TSS, and the technician is prompted to hang up the telephone.

If any communication problems occur during transmission of data to the SEL 86, error messages are displayed and TRAN is exited. The user must rectify the problem using the time-sharing terminal emulation routine, discussed shortly.

2. Conditionally Transmit Data. To conditionally transmit data when TRAN is called from ACQR, a transmit data flag is tested. If set, the data is transmitted as described previously; if not, TRAN returns.

3. Enable Automatic Data Transmission. The transmit data flag is set to enable the automatic transmission of data.

4. Disable Automatic Data Transmission. The automatic data transmission flag is cleared to disable that function.

Search Modem Input Subroutine (SEARCH). Used by TRAN for communication with the SEL 86, the search modem input subroutine examines input from the modem for the occurrence of a given character string. Also, it uses the system clock to allow an error return to be made if the string is not found within a specified time. This permits TRAN to detect a communication problem without hanging the program.

Receive Character String Subroutine (RCVSTR). The receive character string subroutine inputs characters from the modem one at a time using the assembly language modem input function. These characters are concatenated to a string, passed from SEARCH, when RCVSTR is called. An error is returned if the input data buffer for the modem has been overrun or if the acoustic coupler is not ready.

Transmit Character String Subroutine (XMTSTR). To transmit a string to the SEL 86, TRAN calls XMSTR and passes it a variable length character string to be sent. The modem output function is used to transmit the string a character at a time, and a carriage return character is transmitted at the end of the string. An error is returned if the acoustic coupler is not ready.

Time-Sharing Terminal Emulation Subroutine (TTY). The time-sharing terminal emulation routine is called from CMND to allow the operator to communicate directly with another computer system, transparently through the LSI-11. A single character, if any is available, is input from the teletype and transmitted to the modem. Then a character is received from the modem and printed on the teletype. This continues until the operator types a CONTROL Z, then TTY returns to CMND.

BLOCK DATA Section (DEFLT). As discussed in Chapter III, a BLOCK DATA statement is used to define a data area which determines the software configuration of the data acquisition system. System constants are set in DEFLT, and control the operation of FORTRAN routines just examined. The defaults for user-definable system parameters and automatic function flags are also set here so they may be easily modified from one point. Variables which are not system constants or user-definable parameters are

not set in DEFLT, so they will be safely out of reach. The variables that are set in DEFLT are listed in Table III. Not listed are the data and sample time arrays which are declared in DEFLT only so they can be properly dimensioned if the array size is changed.

System Software (Assembly Language)

The assembly language routines for the data acquisition system provide support for the functions performed by the FORTRAN system routines. They are divided into three areas of operation, described in Chapter III: A/D converter, system clock, and modem routines (also see Figs. 6a, 6b, and 6c). Three of the assembly language subroutines mentioned earlier--the synchronization, wait, and initialize modem routines--are not used in the present data acquisition system, so they will not be discussed further. However, source code for these routines, as well as all others, is well-commented.

Subroutines which are called directly from FORTRAN have been written following the compiler's conventions for subroutine linkage, as stated previously in Chapter III. Also, the parameter lists that are passed to the routines are checked for validity to insure they contain the correct number of arguments, and to verify that values are within bounds. Errors that are detected are trapped using the FORTRAN error handler. These precautions will permit the assembly language software to be used in other applications with greater reliability. Furthermore, this software has been designed to be flexible and easy to use in programs which may be developed for other systems.

A/D Converter Subroutines. The A/D converter subroutines support the operation of the A/D converter used with the system, and handle the

Table III

Software System Configuration and Default Data (DEFLT)

Variable	Purpose	Value
CHNMAX	Number of data channels in system	8
MAXSEC	Maximum number of seconds samples can be taken	320 seconds
MAXDAT	Size of data array	1000
MVPBT	A/D converter conversion constant	2.4414 mV/bit
FREQ	Effective system clock frequency	100 Hz
RATE(8)	Sampling rates for channels	User-defined
GAIN(8)	Signal gains for channels	User-defined
OKGAIN(8)	Valid signal gains	1,10,20,50,100, 200,500,1000
NGAINS	Number of valid gains	8
SYNCHN	Synchronization channel	User-defined
SYNGN	Synchronization signal gain	User-defined
SYNMV	Synchronization signal threshold level	User-defined
DELAY	Sampling delay in seconds	User-defined
PLCODE(8)	Plot codes for each channel	User-defined
SPCODES(8, 20)	Special codes for each channel	User-defined
NCODES	Number of special codes for each channel	20
FILNAM	SEL 86 root data file name	User-defined
AQFLAG	Automatic re-acquisition flag	User-defined
PKFLAG	Automatic peak data calculation flag	User-defined
TRFLAG	Automatic data transmission flag	User-defined

compact format in which data is stored. They are:

1. A/D Converter Sampling Subroutine (ADC). The sampling subroutine for the A/D converter is used to sample data on a given system channel, and returns the digitized data, the sample time, and the channel number packed with the data. The flow chart for the algorithm is depicted in Fig. 21.

When the routine is called it is passed a parameter list which is tested, and any errors are trapped. The requested channel number is decremented by one and loaded into the multiplexor address of the A/D converter status register, described in the last chapter. This initiates a conversion. A one was subtracted from the channel number since the FORTRAN programs treat the system channels as being numbered from one to eight, expediting array addressing and DO loop indexing. However, the A/D converter hardware numbers the channels from zero to seven.

After starting a conversion, the done bit of the status register is continuously checked until set, indicating the conversion is complete. Immediately the time of conversion is read from the system clock and stored, then the A/D converter data register is saved. Also, the channel number is packed with the data in the format discussed in Chapter III. The data, sample time, and channel number packed with the data are optionally passed back depending on the configuration of the parameter list. This gives the calling program complete flexibility in determining what data is returned.

2. Unpack Data Subroutine (UNPKCD). To unpack the compact channel number and data format returned by ADC, the unpack data routine is called. Passed the packed data, it returns the channel number offset

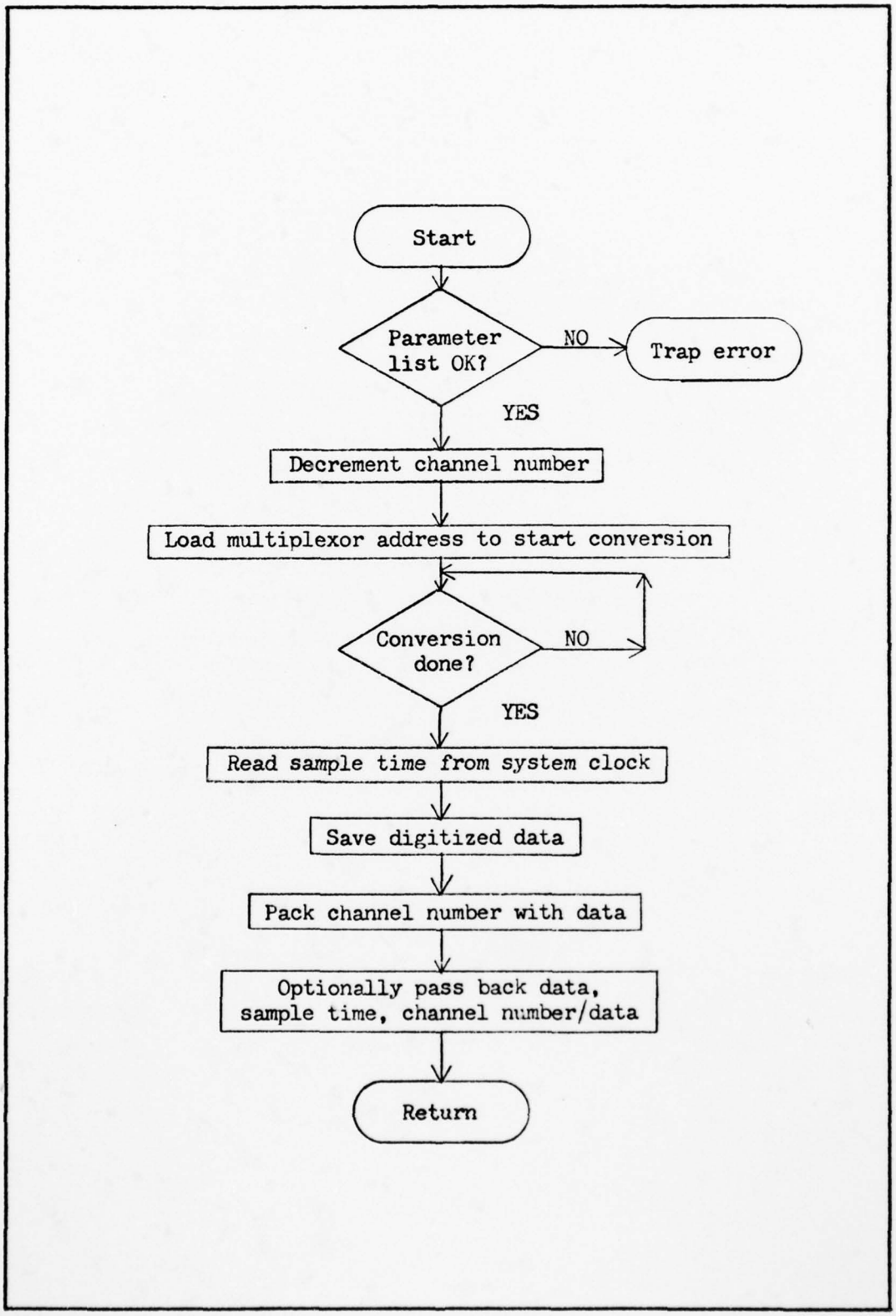


Figure 21. A/D Converter Sampling Subroutine

to between one and eight, and the unpacked A/D converter data.

System Clock Routines. There are two timing requirements which must be fulfilled for the data acquisition system. The first of these is that a time base must be provided for the sampled data which is accurate to 0.01 seconds for tests of five minutes duration. The external clock provided for use with the system operates at 1000 Hz, which would allow an accuracy of 0.0001 seconds. If a 16-bit word is incremented for each clock interrupt the system can keep track of $2^{16} - 1$ (65535) clock pulses. However, this corresponds to a maximum time of about 65 seconds and is insufficient for the 300 seconds required. A multiple clock word could be used but this would complicate handling of times and increase storage requirements. Instead, by ignoring nine out of ten interrupts an effective clock frequency of 100 Hz can be realized. This still maintains the 0.01 second accuracy needed and yields a maximum clock time of 655.35 seconds, or 327.67 seconds if the sign bit of the clock word is ignored.

The second requirement is that timing must be provided so data can be taken at the desired rates, which vary for different channels. This is accomplished by providing a software timer, numbered one to eight, for each channel. An extra timer, number zero, is available for miscellaneous timing functions. Using timers that are automatically counted down during each effective clock interrupt, intervals of up to 327.67 seconds may be measured, accurate to 0.01 seconds. Also, by using flags that can be checked from FORTRAN code to indicate the end of a given interval there is no need for time-consuming subtractions and comparisons of clock times. By automatically resetting the timers for the next interval an addition burden is removed from the system programs.

The clock and timer routines which support these functions are discussed next. They all operate based on an effective clock frequency of 100 Hz, and times are in hundredths of seconds.

1. Clock Interrupt Service Routine (CLKINT). The clock interrupt service routine increments the clock time for the system time base, and services the software interval timers. Fig. 22 shows the basic flow chart for the algorithm.

The interrupt service routine is entered when hardware clock interrupt occurs and the LSI-11 traps to the service routine address in location 100_8. A software switch is tested to see if the system clock is enabled, and if it is not, all interrupts are ignored. If the clock is running only every tenth interrupt is serviced and all others are ignored, as discussed above. For the tenth interrupt the clock time word is incremented. Then each of the enabled interval timers, indicated by a positive timer value, is decremented. If the result is zero an interval expired flag for that timer is incremented, and the timer is automatically reloaded with the value for the next interval. If the timer is not zero then the next timer is serviced; after all timers are checked the interrupt service routine exits.

A data area is used for CLKINT which contains the clock count, clock enable switch, interval timers, values for timer intervals, and interval expired flags. This data is accessed by the other routines that service the system clock and timers. Additionally, the clock time is read by ADC to return the data sampling time. The data area is also organized as a COMMON block (CLKCOM) so that the system clock and timers can be accessed directly from FORTRAN if desired. This is not done by the data acquisition system since, although it decreases efficiency, the subroutine calls

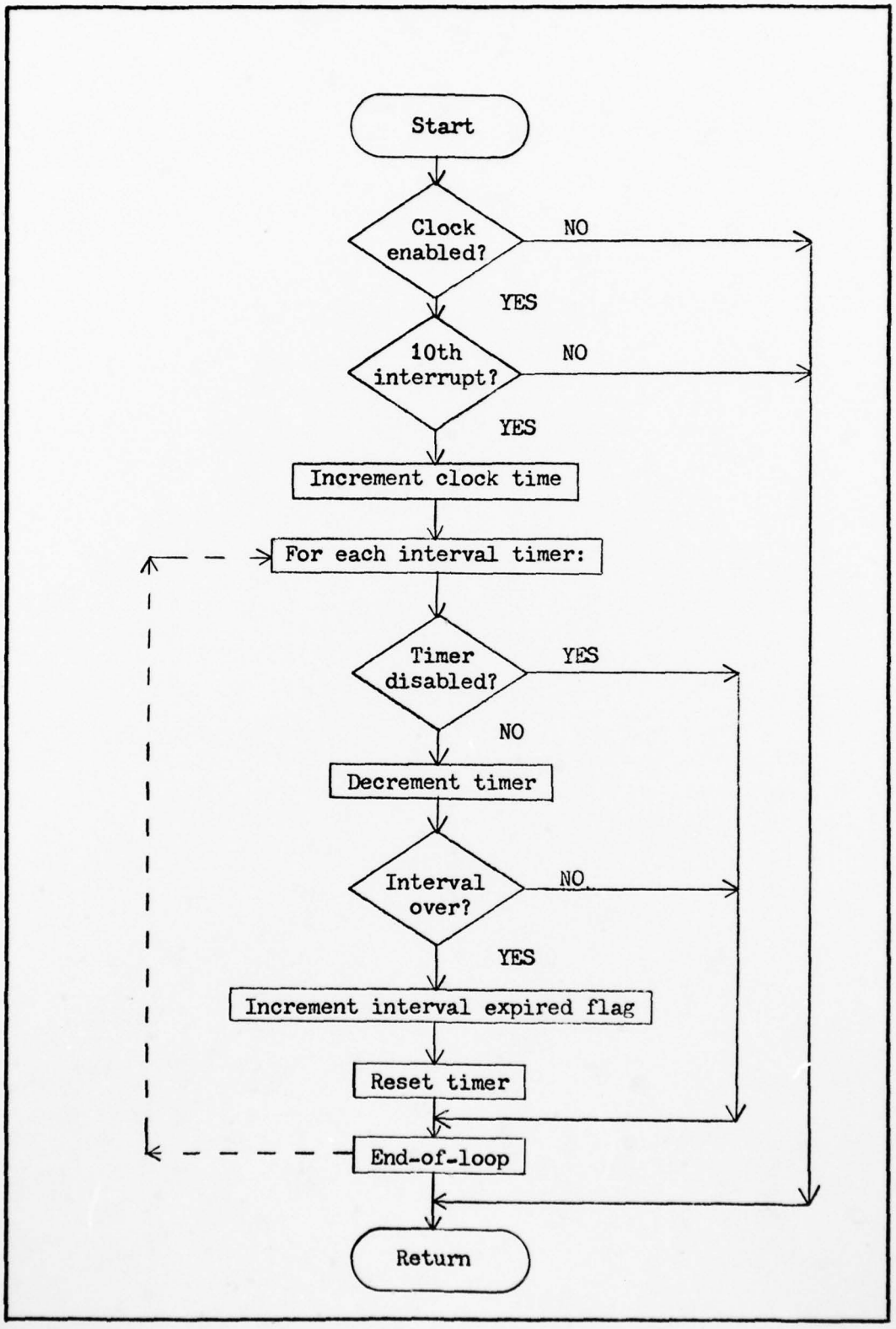


Figure 22. Clock Interrupt Service Routine

used to operate the clock and timers clearly show what is being done, and relieve the user of any requirement to know the internal operation of the system clock.

2. Clock Switch Subroutines (CLKON, CLKOFF). The system clock is enabled and disabled by respectively setting and clearing the software clock switch. This permits the clock to be set and the timers to be loaded while the clock is not r ning, and then all started at the same time.

3. Clock Subroutine (CLOCK). The clock subroutine is used to load the system clock or read the present clock time. The function performed is determined by a flag passed to the routine.

4. Timer Subroutine (TIMER). The timer subroutine is used to set the interval for one of the timers, or to read the time left before an interval expires. When setting a timer the timer interval is stored, the timer is loaded with the new interval value, and the interval expired flag is cleared. Setting the timer with a negative number disables it. A flag controls whether a read or set timer operation is done.

5. Interval Check Function (INTRVL). The interval check function returns the value of the interval expired flag for a given timer. A zero is returned if no interval has elapsed for the timer; else the number of intervals elapsed since the flag was last checked is returned. The interval flag is cleared after being read by this function.

Note that the interval flag may indicate that more than one interval has elapsed. This keeps track of intervals which may be missed by the calling program, and allows intervals longer than 327.67 seconds to be timed since multiple intervals are recorded.

Modem Routines. The modem routines handle I/O through the acoustic coupler modem, enabling the LSI-11 and data acquisition system to communicate with other computers in the network. The routines used for the modem transmitter and receiver operations will be discussed after some brief comments on design implementation.

Interrupt driven I/O is used for the modem to eliminate any need for the FORTRAN code to be concerned with details of modem operation. Also for this reason, the modem I/O routines which interface with FORTRAN are implemented as simple functions to output a character to the modem and input a character from the modem. These functions return values to indicate a successful operation, or a variety of error conditions. Use of the functions is patterned after the FORTRAN library functions to output and input single characters to and from the console terminal (ITTOUR, ITTINR--Ref.10:0-81, 0-79). This will help alleviate any confusion on the part of the programmer using both sets of functions, as is the case for the data acquisition system.

The transmitter and receiver I/O character buffers for the modem are logically organized as circular buffers. Buffer format is depicted in Fig. 23. Pointers are used to indicate the fixed upper and lower bounds for the buffer areas, and to enable buffer wrap-around. Additionally, a buffer input pointer is set to the next free byte position in the buffer, and an output buffer pointer indicates the next available character. Flags are also used to denote the special conditions of buffer empty, full, or overrun.

1. Modem Transmitter Routines. The routines which handle character transmission through the modem perform three functions:

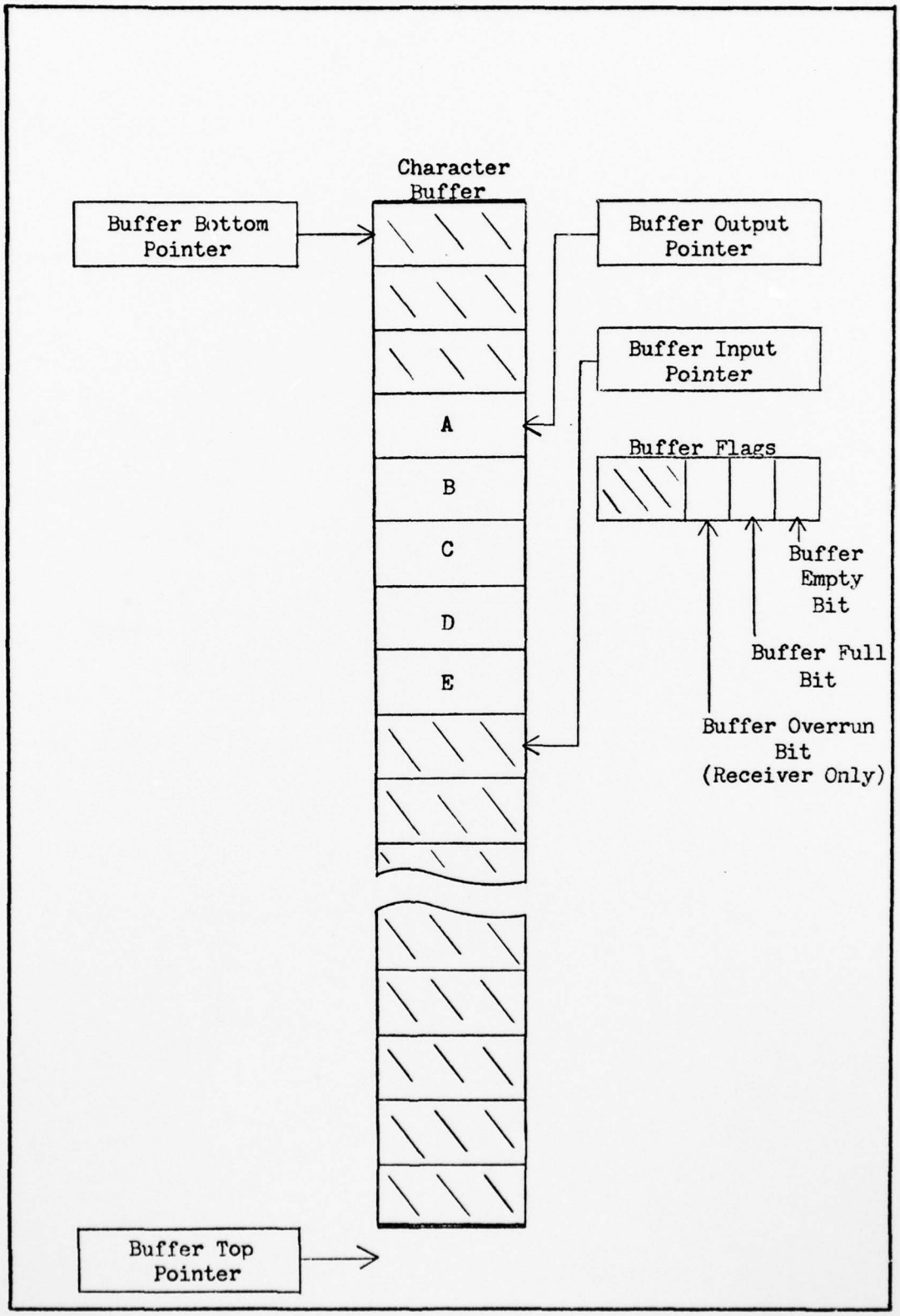


Figure 23. Modem I/O Buffer Format

a) transmit a character to the modem from the output buffer, b) service transmitter interrupts, and c) output a single character from FORTRAN to the output buffer.

a) Transmit Character Subroutine (XMIT). The transmit character subroutine is called to remove a single character from the modem output buffer and transmit it through the acoustic coupler. The character to be sent is placed in the transmitter data register of the serial interface, described in Chapter IV. The buffer pointers and flags are set as needed.

b) Transmitter Interrupt Service Routine (XMTINT). When a transmitter interrupt is received from the modem interface, the transmitter interrupt service routine is entered. If a character is available in the modem output buffer XMIT is called to transmit the next character in the buffer.

c) Modem Output Function (MODOUT). The modem output function is called from FORTRAN to write a single character in the output buffer for transmission to the modem. It is passed a single argument which is the character to be transmitted, or a null argument. If a null argument is used it indicates that no data is to be transmitted, but a test is to be made to determine if the modem is ready. If a character is to be transmitted the test is also made, then the character is placed in the output buffer if room is available and XMIT is called to attempt to output another character. The function returns values to indicate whether the character was successfully placed in the output buffer (0), the output buffer was full (+1), or the modem was not ready (-1).

2. Modem Receiver Routines. The modem receiver routines

handle reception of character data from the modem and perform three functions, similar in operation to those for the modem transmitter.

a) Receive Character Subroutine (RECV). The receive character subroutine removes a character from the receiver data register of the modem serial interface, and stores it in the modem input character buffer.

b) Receiver Interrupt Service Routine (RCVINT). The receiver interrupt service routine is entered when a modem receiver interrupt occurs. If room is available in the input buffer, RECV is called to input a character and store it. An overrun flag is set if the buffer is full.

c) Modem Input Function (MODIN). Called from FORTRAN, the modem input function reads a character from the modem input buffer. It returns a character if one is available, a buffer empty error (-1) if there is no character in the buffer, or a buffer overrun error (0) if data has been lost because of an overrun. The routine also calls RECV to attempt to read a character.

Other Software

Some auxiliary software was written during the development of the data acquisition system. Part of this software consisted of programs written to test the operation of certain aspects of data acquisition system hardware or software, and will be discussed under results in the final chapter. Additionally, some programs were developed to perform specialized tasks for the user. These programs were used to calibrate test facility equipment, including radiometers and slug calorimeters; sample and print data from input channels; and perform calculations to

determine peak lamp flux and total fluence. They demonstrated to the user other capabilities which could be achieved by an automated data acquisition system, and permitted limited use of the system before it became operational.

An absolute loader for the high speed paper tape reader used during data acquisition system software development was also written. It loads object tapes generated by the software development system which are in DEC standard absolute loader format (Ref. 6:G-1 to G-2). The source code is included with the source code for the data acquisition system.

Summary

This chapter has concluded the discussion of the data acquisition system by examining the software which has been developed. The operation of the software, whose organization was described in Chapter III, has been discussed along with some design considerations. The next chapter presents some results, conclusions, and recommendations.

VI. Results, Conclusions, and Recommendations

Introduction

This chapter discusses the results of the design and implementation of the data acquisition system, and states some conclusions drawn. Recommendations for future system improvements are also presented.

Results

The objective of this project, stated in Chapter I, was to develop a LSI-11 based automated data acquisition system for the AFML Thermal Flash Test Facility. The problem was further defined by user specifications in Chapter II. Achieving this objective required the accomplishment of two major tasks: the assembly and interconnection of required hardware, and the design and development of appropriate LSI-11 software. The results of these two tasks will now be discussed, and will include an examination of system tests for both.

Hardware Assembly. The assembly and interconnection of system hardware was successfully accomplished by properly configuring each of the hardware components for the system, and constructing cabling as required. The computer modules -- LSI-11 processor, memory modules, and interface modules -- were each configured by selecting jumpered options, as discussed in Chapter IV. The hardware required to supply the processor control signals was built, and the clock signal line was connected to the LSI-11 bus. Also, the teletype was modified to enable computer control of its low-speed tape reader. Cables were constructed to connect the modem to its interface, the signal conditioning amplifiers to the A/D converter, and the signal sources to the amplifiers. The system backplane

was configured by inserting the A/D converter and computer modules, connecting the two LSI-11 backplanes used, and supplying power to each backplane. The signal amplifiers and the A/D converter were supplied in working condition; however, they have not been calibrated to manufacturer's specifications (Refs. 3,4). Additionally, the hardware necessary to generate the automatic acquisition start signal has not been permanently installed in the test facility control equipment.

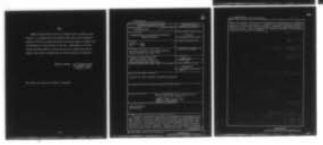
Hardware Testing. The system hardware was tested by observing analog data signals with an oscilliscope to check signal cables and amplifier operation. Some noise problems were observed, but these were corrected by proper shielding. The A/D converter was tested using a program supplied by the manufacturer (Ref. 3:25-28) and using programs developed for software testing, discussed later. Validation of clock and modem operation was also done using programs developed to test each of these components and their accompanying software drivers. These programs additionally served to test the other computer system components--the LSI-11 processor, memory, and teletype. All hardware components for the system have been found to operate satisfactorily.

Software Implementation. The development of the software for the data acquisition system was the second task accomplished. This involved the design and development of the modular FORTRAN routines to perform the major system functions: the command input decoder; the test and plot parameter handlers; the acquisition control and data acquisition subroutines; the peak data calculation subroutine; the data transmission routine with its attendant subroutines; and the time-sharing terminal emulator. Assembly language programs were also developed to provide

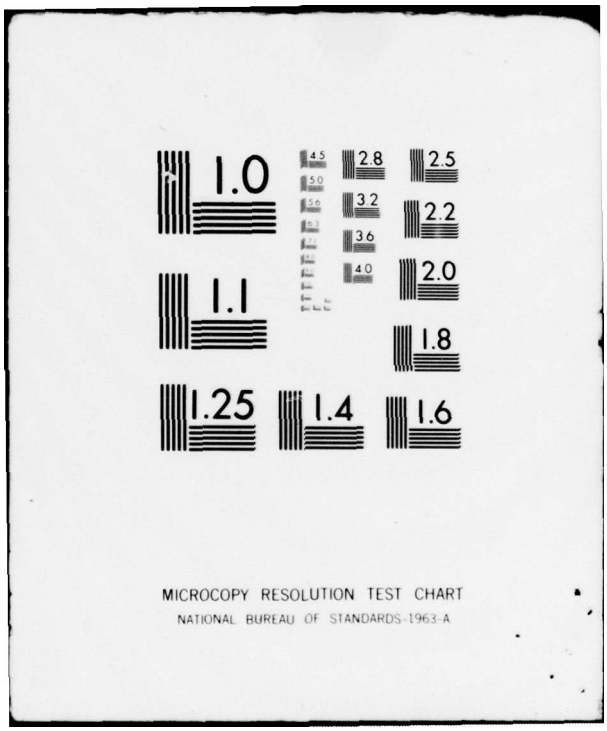
AD-A055 466 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING.(U)
MAR 78 R G RUSHE

UNCLASSIFIED AFIT/GEP/EE/78-1 NL

2 OF 2
AD
A055466



END
DATE
FILMED
8 -78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

support for the FORTRAN system routines. These were the A/D converter sampling and unpack data subroutines; the system clock and interval timer routines; and the modem receiver and transmitter routines.

All of this software has been implemented and tested, as discussed in the next section. The total data acquisition program, however, is larger than the 8K words originally desired to be the maximum size. This can be traced primarily to the extensive use of FORTRAN in the system, a constraint imposed by the user. Other contributing factors are the general and flexible nature of the software, and the error checking and interactive communications used to simplify system operation for the user. The system has still been implemented with the addition of 8K words of memory, for a total of 16K, which is sufficient for the final program size of about 14K.

Software Testing. Since the software for the data acquisition system is modular, testing of system software was achieved by testing each module either separately, or as it was integrated with previously validated system components. This testing served to insure that each module performed its assigned operations, and that all functional system requirements were individually met. Primary testing of the FORTRAN modules was done on the development system to make it easier to effect any required corrections. The assembly language modules had to be individually tested on the data acquisition itself since they require certain hardware components for operation --the A/D converter, clock, and modem --which were not available on the development system. Finally, all tested modules were integrated and the operation of the data acquisition system as a whole was verified. The testing procedure and results will now be discussed.

1. FORTRAN Module Tests. The FORTRAN modules were tested as they were developed, starting with the command decoder and continuing through each of the major submodules. As the operation of each was verified and corrected as needed, it was integrated with the previously tested sections of the system software. For the FORTRAN routines which require the assembly language functions, short subroutines were written to emulate those functions on the development system.

The command decoder was the first of the FORTRAN modules to be written. It was tested by setting up dummy subroutines for each of the lower modules and verifying that these were called in response to commands which were entered. Then as each of the submodules was developed it replaced the corresponding dummy subroutine, and was called from the command decoder to be tested individually.

The test and plot parameter handlers were checked by entering sets of parameters, containing valid and invalid data, and observing that the routines operated correctly and detected any errors in input such as values out of bounds or character instead of numeric data. Printing of the parameters using the print routines verified that the stored values of the parameters were correct, and that the print routines themselves were working properly.

The acquisition control subroutine was tested by also verifying that it handled input data correctly, and by observing that it called the data acquisition, peak data calculation, and data transmission subroutines as required. These were still only implemented as dummy routines.

To test the data acquisition subroutine on the development system, the functions of the A/D converter and clock routines were emulated using

simple subroutines to generate data. An additional section of code was inserted so that data thus 'acquired' could be printed out. The data acquisition subroutine was then tested by calling it from the acquisition control routine and verifying proper operation. The data acquisition routine was also tested with the required assembly language routines on the data acquisition system to insure that automatic acquisition synchronization functioned correctly, and to check operation with real data prior to the system integration test.

The peak data calculation routine was checked by loading the data array with a set of test data, and then verifying that the correct peak data was retrieved when the routine was called.

In order to test the operation of the data transmission routine on the development system a special subroutine was written to allow the console terminal to send and receive all data normally input and output through the modem. The data transmission routine was then tested by typing in the messages that are transmitted by TSS from the SEL 86. The response of the transmission routine was observed to check that it was correctly sending the required TSS commands, parameters, and test data. The operation of the search string, receive string, and transmit string subroutines had been verified prior to this by individually testing each routine with a simple driver.

The teletype emulation routine was also tested using the special subroutine developed to test the data transmission routine.

Now that each of the FORTRAN modules had been tested, corrected, and integrated into the system, the program as a whole was exercised to the extent possible on the development system. Minor deficiencies were

corrected, and the data acquisition system FORTRAN software operated satisfactorily.

2. Assembly Language Module Tests. The assembly language modules were initially tested separately from the FORTRAN modules just discussed, before integrating them into the system. This involved writing short FORTRAN test drivers for each group--the A/D converter, system clock, and modem routines--and loading them into the data acquisition system to verify correct operation.

The A/D converter routines were tested by having the sampling subroutine sample a data channel, with a known signal level, and then return the digitized data in both packed and normal format. The unpack data subroutine was used to unpack the packed data, which was then compared with the normal data returned by the sampling routine. Both of these were multiplied by the conversion constant and the resulting voltage figures were compared with the known input voltage to verify proper operation. This test sequence also served to show that the A/D converter hardware was working correctly.

System clock and interval timer operation were validated by setting the clock and timers to certain values and then measuring elapsed time against an external line clock. The test program was allowed to run for several hours to insure there was no drift in the system clock because of hardware or software problems.

The modem routines were tested using a program similar to the FORTRAN time-sharing terminal emulation subroutine. This allowed verification of the proper operation of the modem routines, as well as the acoustic coupler, by testing communication with the time-sharing systems of the

SEL 86 and CDC 6600.

After any errors uncovered were corrected in the assembly language modules, they functioned properly and were ready to be integrated into the final system.

3. Integrated System Test. All of the FORTRAN and assembly language modules, each tested previously, were integrated and loaded into the data acquisition system for an overall test of system operation. A few minor bugs were discovered and most were corrected with no difficulty. However, one aspect of system operation still experiencing problems is the transmission of data to the SEL 86. Communication with TSS by the data transmission routine has not yet been completely successful. Since the transmission routine operated correctly when manually tested on the development system, there is probably some undiscovered variability in the rather informal communications protocol used.

In addition to the problems with the transmission module, a complete operational test of the system cannot be achieved until the user has developed the plotting programs on the CDC 6600 to accept the data gathered by the data acquisition system.

Conclusions

The automated data acquisition system which has been developed represents a significant improvement over the original system. It increases the data acquisition capabilities of the Thermal Flash Test Facility by permitting more channels to be sampled at higher rates and with greater accuracy than was possible before. The automatic transmission of data will eliminate the need to physically transport the data, and will increase test facility productivity by decreasing test turn-around time.

The general nature of the system will preclude it from becoming obsolete; and in fact, the system should serve as the nucleus for future expansion. Furthermore, much of the software developed and lessons learned will be applicable to similar systems developed at the Materials Laboratory.

Recommendations

There are a number of recommendations which can be made for completing the work done to date, and for future system improvements. Near-term actions which must be accomplished before the system can be considered fully operational are:

1. Calibrate the signal conditioning amplifiers and the A/D converter.
2. Finish installing the hardware necessary to generate the automatic acquisition synchronization signal.
3. Complete the debugging and testing of the data transmission routine.
4. Develop the CDC 6600 plotting programs to handle the data gathered and transmitted by the system.
5. Perform testing for complete system operation--from analog data in to plots out.

The long-range recommendations for system improvements should increase the usefulness and performance of the data acquisition system; moreover, some are applicable to similar systems which may be developed. They are:

1. Design additional hardware and software for the data acquisition system to enable the LSI-11 to control the operation of test facility equipment.

2. Modify or develop new SEL 86 software to improve the data transmission capabilities of the link with the LSI-11. This might include the design of a formal protocol with error checking features and less communications overhead.
3. Increase the baud rate of the communications link to the SEL 86. This will improve data acquisition system operation even if no new protocol is developed by decreasing the time required to transmit data.
4. Eliminate paper tape as the object code medium, possibly using the SEL 86 as a program repository for the data acquisition system.

Bibliography

1. A242/AD342 Acoustic Couplers Operating Manual. Sunnyvale, California: Anderson Jacobson, Inc.

2. Brignel, John E. and Godfrey M. Rhodes. Laboratory On-Line Computing. London: International Textbook Company Limited, 1975.

3. Instruction Manual for ADAC Corporation Model 600-LSI-11 Data Acquisition and Control System. Woburn, Massachusetts: ADAC Corporation.

4. Instruction Manual for the 687 Instrumentation Amplifier. San Diego: Ectron Corporation.

5. Millet, E. J. "Digital Techniques in Laboratory Automation." Journal of Physics E: Scientific Instruments, 9: 794-802 (1976).

6. Minicomputer Handbook. Maynard, Massachusetts: Digital Equipment Corporation, 1976.

7. Morrison, Ralph. Grounding and Shielding Techniques in Instrumentation. New York: John Wiley and Sons, Inc., 1967.

8. PDP-11 FORTRAN Language Reference Manual. DEC-11-LFLRA-C-D. Maynard, Massachusetts: Digital Equipment Corporation, 1975.

9. RT-11/RSTS/D FORTRAN IV User's Guide. DEC-11-LRRUA-A-D. Maynard, Massachusetts: Digital Equipment Corporation, 1975.

10. RT-11 System Reference Manual. DEC-11-ORUGA-C-D, DN1, DN2. Maynard, Massachusetts: Digital Equipment Corporation, 1976.

11. Rushe, Randall G. Source Code for Thermal Flash Test Facility Data Acquisition System. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1978.

12. Santucci, Dennis. "Maneuvering for Top Speed and High Accuracy in Data Acquisition." Electronics, 48: 115-119 (November 27, 1975).

13. Sebern, Mark J. "A Minicomputer - Compatible Microcomputer System: the DEC LSI-11." Proceedings of the IEEE, 64: 881-888 (June 1976).

14. Servais, R. A., B. H. Wilt, and N. J. Olson. Tri-Service Thermal Radiation Test Facility: Test Procedures Handbook. UDRI-TR-77-28. Dayton, Ohio: University of Dayton Research Institute, 1977.

15. Terminal Support Subsystem Reference Manual. Ft. Lauderdale, Florida: Systems Engineering Laboratories, 1977.

Appendix A

Hardware Documentation

Introduction

This appendix contains additional detailed hardware documentation for the data acquisition system. Included are pin assignments and hardware module configurations.

Table A1

Amplifier - A/D Converter Connector Pin Assignments

Ectron Output Connector (J17)		ADAC Input Connector (J1)	
Pin	Channel	Pin	Channel
1	1 high	16	0A
19	1 low	33	0B
2	2 high	15	1A
20	2 low	32	1B
3	3 high	14	2A
21	3 low	31	2B
4	4 high	13	3A
22	4 low	30	3B
5	5 high	12	4A
23	5 low	29	4B
6	6 high	11	5A
24	6 low	28	5B
7	7 high	10	6A
25	7 low	27	6B
8	8 high	9	7A
26	8 low	26	7B
-	Ground	49	Power Return

Table A2

Ectron Amplifier Input Connector Pin Assignments

Pin (Connector J1)	Signal (Channel 1)
C	low
B	high
E	shield guard
F,G	cold junction sense (UTA only)

Repeated on connectors J2-J8 for channels 2-8.

Table A3

KD11-F Processor Module Configuration

Jumper	Function
$\overline{W1}$, W2	bank 0 addresses selected for resident memory (000000_8-017777_8)
$\overline{W3}$	event line interrupt enabled
$\overline{W4}$	memory refresh enabled
W5, $\overline{W6}$	ODT power-up mode

Table A4

MSV 11-B Memory Module Configurations

	Jumper	Function
Module 1:	W1, W2, $\overline{W3}$	bank 1 addresses selected (020000_8 - 037777_8)
	$\overline{W4}$	reply during refresh enabled
Module 2:	W1, $\overline{W2}$, W3	bank 2 addresses selected (040000_8 - 057777_8)
	W4	reply during refresh disabled
Module 3:	W1, $\overline{W2}$, $\overline{W3}$	bank 3 addresses selected (060000_8 - 077777_8)
	W4	reply during refresh disabled

Table A5

Teletype DLV11 Serial Interface Module Configuration

Jumper	Function
CL1, CL2, CL3, CL4, $\overline{\text{EIA}}$	20mA, active current loop operation
$\overline{\text{FR0}}$, $\overline{\text{FR1}}$, $\overline{\text{FR2}}$, $\overline{\text{FR3}}$	110 baud
$\overline{\text{NP}}$	no parity
$\overline{\text{2SB}}$	two stop bits
$\overline{\text{NB1}}$, $\overline{\text{NB2}}$	eight data bits
$\overline{\text{PEV}}$	even parity (don't care)
FEH	framing error halt
$\overline{\text{A3}}$, $\overline{\text{A4}}$, $\overline{\text{A5}}$, $\overline{\text{A6}}$, $\overline{\text{A7}}$, $\overline{\text{A8}}$, $\overline{\text{A9}}$, $\overline{\text{A10}}$, $\overline{\text{A11}}$, $\overline{\text{A12}}$	device address 177560_8 selected
V3, $\overline{\text{V4}}$, $\overline{\text{V5}}$, V6, V7	interrupt vector address 60_8 selected

Table A6

Modem DLV11 Serial Interface Module Configuration

Jumper	Function
EIA, $\overline{\text{CL1}}$, $\overline{\text{CL2}}$, $\overline{\text{CL3}}$, $\overline{\text{CL4}}$	EIA, RS232C operation
$\overline{\text{FR0}}$, FR1, $\overline{\text{FR2}}$, $\overline{\text{FR3}}$	300 baud
$\overline{\text{NP}}$	no parity
2SB	one step bit
$\overline{\text{NB1}}$, $\overline{\text{NB2}}$	eight data bits
$\overline{\text{PEV}}$	even parity (don't care)
$\overline{\text{FEH}}$	no framing error halt
$\overline{\text{A3}}$, A4, A5, A6, $\overline{\text{A7}}$, $\overline{\text{A8}}$, $\overline{\text{A9}}$, A10, $\overline{\text{A11}}$, $\overline{\text{A12}}$	device address 175610_8 selected
V3, V4, V5, $\overline{\text{V6}}$, $\overline{\text{V7}}$	interrupt vector address 300_8 selected

Table A7

Acoustic Coupler - Modem Interface Connector Pin Assignments

Acoustic Coupler Pin	Signal	DLV11 Interface Pin
1	protective ground	A, VV
2	transmitted data	F
3	received data	J
4	request to send	V
5	clear to send	T
6	data set ready	Z
7	signal ground	B, UU
8	data carrier detector	BB
20	data terminal ready	DD
	TTL data in	M jumpered to E

Vita

Randall George Rushe was born on 19 December 1954 in Columbia, South Carolina. He graduated from the dependent high school at RAF Lakenheath, England, in 1972 and received his Bachelor of Science degree in Physics from the University of South Carolina in May 1976. Commissioned in the USAF through the ROTC program, he entered active duty in August 1976 when assigned to the School of Engineering, Air Force Institute of Technology.

Permanent address: 147 Alexandria Avenue
West Columbia, South
Carolina 29169

This thesis was typed by Ms. Sherry L. Markiewicz

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GEP/EE/78-1	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A MICROCOMPUTER DATA ACQUISITION SYSTEM FOR MATERIALS TESTING		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Randall G. Rushe 2Lt USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Computer Activities Office Air Force Materials Laboratory Wright-Patterson AFB, Ohio 45433		12. REPORT DATE March 1978 13. NUMBER OF PAGES 110
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Data Acquisition
Microcomputer

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Laboratory computers for automated data acquisition are becoming increasingly common. This report summarizes the development of a microcomputer-based data acquisition system for the Air Force Materials Laboratory Thermal Flash Test Facility. The system is based on a Digital Equipment Corporation LSI-11 microcomputer and acquires multiple channels of data from nuclear simulation experiments on aircraft components. Comprising the lowest level of a rudimentary hierarchial network, the system transmits the data to a larger minicomputer

(CONTINUED)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

system for storage, from where it is subsequently retransmitted to a mainframe system for reduction and plotting. Software for the data acquisition system is modularly structured and written primarily in FORTRAN to facilitate modifications by the user. Basic hardware for the system is discussed, and the program algorithms and structure are examined.